



# 穿越 计算机的迷雾

李忠◎著

- 你苦恼于枯燥的计算机知识吗？
- 你知道计算机是怎么工作的吗？
- 你真正了解计算机吗？
- 这本通俗易懂、平易近人、妙趣横生的书，会带你穿越萦绕在脑海中的种种迷雾，为你揭开蒙在计算机上的层层神秘面纱。



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

# 目录

- [第1章 了解计算机，要从电开始](#)
- [第2章 用电来表示数](#)
- [第3章 怎样才能让机器做加法](#)
- [第4章 电子计算机发明的前夜](#)
- [第5章 从逻辑学到逻辑电路](#)
- [第6章 加法机的诞生](#)
- [第7章 会变魔术的触发器](#)
- [第8章 学生时代的走马灯](#)
- [第9章 计算机时候的开路先锋](#)
- [第10章 用机器做一连串的加法](#)
- [第11章 全自动加法计算机](#)
- [第12章 现代的通用计算机](#)
- [第13章 集成电路时代](#)
- [第14章 核心与外部设备间的接口](#)
- [第15章 计算机的启动过程和操作系统](#)
- [第16章 办公、娱乐和程序设计](#)

# 第1章 了解计算机，要从电开始

这是一本讲述计算机奥秘的书。计算机的工作需要电，只有打开电源计算机有运行的能源，这意味着要想知道计算机是怎么造出来的，又是如何工作的，必须了解电。先有了电，于是才有了电灯，电灯为什么要这样构造？那是因为我们发现只有这样构造，电才能为我们发光。不了解电，永远也不会明白电子计算机为什么非得是这个样子，它到底是怎么工作的。

## 1.1有的东西能导电，而有的则不能

要输送电，通常只能借助于电线，根据我们在日常生活中得到的经验，电只能通过金、银、铜、铝、铁等物质来传输，而对于干燥的木头、纸张、塑料、陶瓷等物质则不行。这基本上属于常识

能够导电的物质叫导体，通常情况下不导电的物质叫绝缘体，但也不总量这样，有时候，像木头这样的东西，在被雨水淋湿了之后也会导电。可见导体和绝缘体之间的界限并不是十分绝对的。由于人体的组成大部分是水，所以人体也能导电

尽管看不见，但几乎所有人都会“感觉”电像非常小的颗粒，它们可以在导体中游走。原则上这种认识并没有错，从18世纪以来科学家就在思考和研究这件事情，并把这种非常微小的粒子叫电荷

你可以随意想象电荷在物体里穿行的样子，比如可能认为它是一束束地在导体里游动，导体有空隙而绝缘体没有，所以绝缘体不能导电。但需要指出的是实际情况并非如此。要清楚这一点，需要了解世间万物的微观构造

## 1.2 电的老家是原子

理论上任何物质都可以无限地分割的。比如一根铅笔，从中间折断，分成的两段再从中间折断，可以不停地分割下去，因为不管每一段多么短小，它依然是实际存在的物质。

所有物质都可以无限分割，这只是一种理论上的假设，但没有谁能够把物质无休止地分割下去

“分割”是一种非常粗暴的行为，常常带有破坏性。与分割不同，古代就有人通过观察发现很多物质自然能分解成细微的组成部分，比如空气，尽管看不见，但当有风时，能感受到它，所有古人认为它必定是以非常微小的颗粒组成的。比如糖块，平时它是固体，即使碾成粉末，也是看得到的小颗粒。但如果把它丢进水里，就会消失，当水蒸发后，又会出现糖的颗粒，这意味着，像糖这样的东西应该是由非常细小的微粒组成的，这些微粒太小，人眼看不到

事实上，不仅仅是空气、糖，世界上所有物质都是由肉眼看不到的微粒组成的。与分割不同，“组成”意味着某种东西是由另外一些东西相互结合在一起形成的

那么组成所有物质的最小微粒是什么呢？物质都是由原子构成的，不同的物质，构成它的原子也不同。

这难道是结论吗？

在古代没有原子的概念，但只要思想是一样的，采用什么表达方式和什么术语是无所谓的。中国，这种思想可追溯到春秋战国时代；在西方，第一个具有这种思想的是古希腊人德谟克里特

德谟克里特关于原子的思想里，整个世界只有两样东西：原子和除了原子之外的“什么也没有”（虚空），原子是不可再分的，它组成了世间万物。原子在数量上是无限多的，万物之所以各不相同，是因为组成它们的原子在数量、形态和排列方式上存在着差异。德谟克里特关于原子的思想在很大程度上是错误的，但他的朴素的原子思想与现代的科学发现比较接近。

科学的原子理论是在18世纪时由一个名叫约翰·道尔顿（John Dalton）的人完成的。道尔顿的故乡是英国人，他1766年出生于一个清贫的织布工家庭，唯一的优势就是 – 用一些传记作家的话来说 – 从小就聪明过人，12岁就当上了当地一所小学的校长。

道尔顿患有色盲症（色盲症也叫道尔顿症），一般来说，人类总是倾向于研究发生在自己身上的各种怪异之事，于是道尔顿也就成了研究色盲的第一个人。

1808年，道尔顿写了一本名为《化学哲学的新体系》的书，在这本书里说明了差不多是现代概念的原子。没有人能创造原子，它一直就存在着，从世界开始的那一天。

“创造毁灭一个氢原子，也许就像向太阳系引进一颗新的行星或毁灭一颗业已存在的行星那样不可能。”

很显然，这个五彩缤纷的世界不可能只用一种原子组成。道尔顿研究了不同类型原子之间的相对大小，以及它们各自的性质和相互之间结合的方法，尽管在他那个时代，已知的原子类型很少，比如，他当时认为我们平时喝的水（无论一碗还是一滴）是由氢原子和氧原子按1：7的比例构成的，这个比例实际上是2：1。

从世界存在的那个时候原子就一直存在。原子有多大呢？大约等于1mm的1/10000000（千万分之一）。如果将一个苹果放大到地球那么大，那么一个原子相当于一个苹果的大小。

尽管公认原子是存在的，但始终还是没有办法看见原子。人们承认它，只是因为越来越多的实验表明它肯定是存在的。一开始，人们觉得原子可能是方的，像砌墙用的砖。听起来似乎有些道理，因为它能很好地解释为什么像金属和钻石这样的东西竟然有那么高的硬度。然而更多的科学家则倾向于认为原子更像一个实心球—一个密度很大的球体。

到20世纪初，人们已经普遍知道原子并不实心球，它实际上还具有更小的组成部分。原子内部绝大部分都是空的，在原子中央有一个非常微小的核心，由数量不等的质子和中子聚集在一起组成的，称为原子核。在原子核的外面，围绕着核外电子，简称电子。

对于原子内部的世界是什么样子，大多数时间我们只能依靠想象，这是一件遗憾的事情。但就算能观察原子的内部，也会觉得没什么意思：原子内部很空旷，尽管里面有一个原子核和一些核外电子，但这些东西对原子内部的空间来说太微不足道了。

通常一个原子不同于另一个原子的原因，是原子核内的质子数不同。比如，组成氢气的氢原子只有一个质子，是所有原子中最简单的；铁原子有56个质子。目前已知的原子有一百多种，这就意味着就最复杂的原子来说，它的质子数会达到一百多个

不同的原子具有不同的性质和特点，用一本科普书的话说，“质子数决定了原子的身份，电子数则决定了原子的性情”。

## 1.3 为什么有些东西可以导电

原子是怎样组成物质的呢？

没有用来把原子粘合在一起的胶水，原子在形成物质时，必须依靠共用外层电子的方法来抱成团。一个原子会有一个以上的邻居，而它的邻居们也一样有自己的邻居，当共用外层电子发生的时候，每个原子外层的电子会定期到别的原子那里待一会儿。这有点像两个无聊的邻居，今天你到我家住，我到你家住，明天再换回来，就这样不停地折腾。由于电子和原子核之间的引力作用，所以当电子在原子之间共用时，电子充当了原子间的黏合剂。

原子之间的黏合剂是它们各自最外层的电子，具体方式就是电子共用。电子共用通常只发生在相邻的原子间，但这不是任意的，不同的原子，也会在“邻居”的类型和数量上有所挑剔，就像俗话说的“人以群分，物以类聚”

从大的、看得见的宏观层面来说，物质分为两大类，第一类物质的共同特点是由同一种原子结合而成，比如金、银、铜等，它们分别都是由相应的金、银、铜原子组成的。第二类物质，其组成方式比第一类物质的组成复杂，例如，水是由两种东西结合而成：氢和氧。氢和氧平时是气体，分别由氢原子和氧原子组成，但当氢气和氧气混合起来燃烧时，氢原子和氧原子就会通过共用电子形成水。再比如我们平时吃的盐，它居然是由两种极其危险的物质构成的——钠和氯，所以食盐在化学上称为氯化钠。钠是一种金属，银白色可以导电。是金属却非常柔软，可以用小刀切成片

原子之间通过共用电子来形成各种各样的物质，这并不是一件容易的事。如果不是这样的话，你穿的衣服会和皮肤慢慢融合；坐在椅子上，椅子会长在屁股上……不同类型的原子，核外电子离原子核的远近不同，原子最外层的电子数也不同，这使得原子有不同的性格特性。原子喜欢什么样的邻居，能与邻居结成哪种形式的关系都是不同的。有时候，很容易形成伙伴关系，有时则需要很高的温度或很高的压力。总之，原子间能够形成什么样的伙伴关系意味着它们想要那样，那样对它们来说最自然、最合适的。

正常情况下，电子不会无缘无故地从原子里跑出来，因为原子是很稳定的，而且，对大多数物质来说，电子的共用总是在相邻的原子之间进行，这种情况下，电子从一个地方到另一个很远的地方去串门是非常困难的，这就是我们通常说的绝缘体。

与绝缘体相反，在导体里，通常共用的电子都不太老实，在自己的

位置上待一会儿就觉得乏味，于是就溜号。如果别的地方正好有个空位置（当然是另一个不老实的家伙溜走留下的），它就迫不及待地跑过去，在这些物质里，几乎充满了这种不负责任的家伙，可以想象这些物质中电子跑来跑去

在经典物理学里，这些不负责任的家伙名字叫自由电子。它们的存在是导体能够导电的根本原因。换句话说，当把一根导线接到电源插座上时，电压推动导线里的自由电子朝着一个方向前进时，“导电”这个过程就发生了



## 1.4 电流是怎样形成的

在导体中，当电子们像马路上的汽车一样，朝着一个方向持续不断地前进时，就形成了所谓的“电流”。电流的速度很快，每秒30万千米，和光速一样。这么看来电子的运动速度真快！错了，这可不是电子的移动速度，每秒30万千米的移动让人觉得很不可思议，更何况自由电子名义上很“自由”，但毕竟受原子核的约束，而且它是在原子的丛林中移动，难免还要磕碰

电子的移动速度其实很慢，每秒移动的距离一般不到1mm，比蜗牛都慢。但当导体两端的电子同时开始移动时（就像正在行军的士兵们），我们就觉得电子好像真的在一瞬间从一头到达了另一头

感觉到电流的速度是一瞬间，或者认为它快到不需要用“速度”来衡量，这只是人类的一种普遍的错觉，因为我们无法制造出一根足够长的电线，如果真有一根从地球到太阳之间距离那么长的电线，那么当你在地球上接通电源，差不多要在8min，太阳那一端的灯泡才能亮起来

电流不是自发形成的，一根扔在角落的电线里面是不会有电流的，要形成电流，最简单的方法就是找一节电池，一个小灯泡，一根电线，并把它们按图1.1所示那样连接起来



图1.1 能让灯泡发光的装置

灯泡的一端接电池正极，另一端通过电线接负极

除了这个实验外我们还知道，所有的电器只有在发电厂开工时才能工作，这意味着而且看上去的确很像 – 电池或发电厂会制造电子，而且会源源不断地把造出来的电子送出来，电流就是这样形成的。在灯泡或其他用电的东西那里，电子被消耗掉，也许是消失了，总之变成了光、热、使轮子旋转的动力，等等。

这是真的吗？

如果这是真的，那么发电厂必须找一个大瓶子，将大量的电子灌进去，然后再用一根电线插到瓶子里，电线的另一头则通向千家万户。这有点像液化气站，在那里贮存了大量液化气，当要用时打开阀门，液化气就来了

真是个好主意，可是到哪里找这么多电子呢？要知道，这可不是小数目，而且，电子可不是萤火虫，能够随便说逮就逮到了，这事儿从来没有人能办到，永远也办不成，除非“直到虾学会吹口哨，或者没有镜子能看到自己的耳朵”

没错儿，这种想法确实过于天真了，而且会让科学家们很生气。要想真正了解电流的原因，需要深入到电池的内部。还是图1.1那个例子，当这个装置开始工作，灯泡开始发光时，电池的作用就像内部安了一个泵（和水泵一样），它促使整个线路中的电子像水流一样不停地循环流动。我们用图1.2中的小白点来代表电子。



图1.2 电流的成因

很清楚，电子不是凭空产生的，而“发电”也不是制造电子，说到底，是让导体中原有的电子循环流动，有几种方法可以让电子们运动起来形成电流，最常用的方法是建造大型发电厂，在那里，工程师们想办法借助于水流或蒸汽的力量驱赶电子，让它们循环往复地流动，这种方法参与的电子多，电流也很强大

在原子的层面，当“电泵”开始工作时，从电线内一边的原子那里夺走自由电子，并将它们送给电线另一边的原子，这样，失去电子的原子很着急，而得到电子的原子也不会因为自己的电子多了而高兴，它们都急切地想要找回电子或扔掉包袱而重新达到稳定状态，在这种情况下，用物理学家的话说“电压产生了！”

电压是一种吸引力，是由于失去电子和希望重新得到电子而引起

的。这是比较抽象的概念。当然也可以把它想象成一种压力，当把一桶水提到高处时，它就具备了流动的可能性。

电压的存在是导致电流产生的原因，在图1.2中，由于整个电路是处处连通的（灯泡其实也只是一段能发光的导线），所以在“电泵”的作用下，电子在整个电路中循环流动。一个原子被迫丢掉原子后，它马上又从别处得到电子，然后不断重复这个过程，除非“电泵”停止工作，换句话说，除非电压不复存在

像大型发电厂里的“电泵”一样，能够产生电压的装置称为电源。如果要严格一些来说的话，电源的作用是产生持续的电压和电流。注意“持续”两个字，这是很重要的。否则接通电源，灯泡只闪一下就灭了

另一种电源是电池，这是我们都熟悉的东西，它和前面所讲的发电原理基本一样，但稍有不同。电池也不制造电子，只是把电子从电池的一端搬到另一端。电池有正极和负极，我们被告知电从电池的正极出来，然后流回负极，实际上，真实的情况是电子的运动方向是从负极出来，流回正极。电学的先驱们犯了一个小错，但无伤大雅，所以就一直沿用下来了

当然，能够产生电压和电流的东西很多，但不一定持久，所以用来作为电源可能不会很理想，例如静电，静电也是因为有些原子失去电子而另一些原子得到电子，如果有机会，这些原子会迫不及待地重新达到稳定状态，导致放电现象。另外，雷电也是电。当天空中云层和地面分别因为得到电子和失去电子而是带上静电时，如果时机合适，放电过程就在雷声中开始了，而且能看到明亮的闪电。人们直到最近几个世纪才认识到天上的雷电也是电

# 富兰克林的风筝实验？

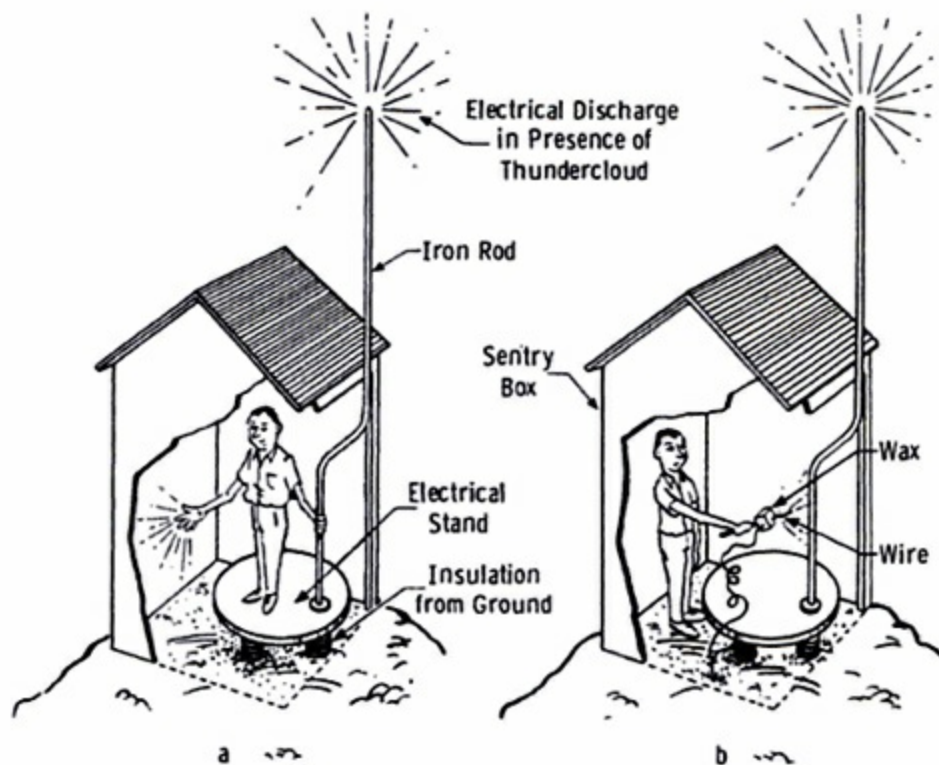
据说本杰明·富兰克林（Benjamin Franklin）将一把铜钥匙系在风筝线的末端，然后在雷雨天时放飞风筝，当风筝升入雷雨云层，闪电在风筝附近闪烁，雷声隆隆，一道闪电掠过，风筝线上有一小段直立起来，像被一种看不见的力移动着。富兰克林突然觉得他的手有麻木的感觉，就把手指靠近铜钥匙，铜钥匙上射出一串火花。富兰克林大叫一声，赶紧把手远离钥匙，喊道“威廉！我受到电击了！现在可以证明，闪电就是电！”

## 真相

富兰克林的成就实在数不胜数，他参加起草了《独立宣言》和美国宪法，担任过州长，是美国历史上第一位驻外大使，出版了费城第一份报纸《宾夕法尼亚报》、美国第一本医学专著、第一部小说及其他很多畅销书，在多个科学领域也有很多贡献（他甚至成了财富的象征）

富兰克林生前身后流传着很多故事，这个带电风筝的故事就是其中之一。作为流传已久的经典传说，很多美国人深信不疑。在国内，作为语文教科书里的“老段子”，同样人尽皆知，可是这到底是神话还是真有其事呢？

富兰克林是第一个提出用实验来证明天空中的闪电就是电的科学家。但是第一个付诸于实践的却是法国科学家，1752年5月，乔治·路易·勒克莱尔（George Louis Leclerc，布丰伯爵）像下图中那样观察到了铁棒上的火花，不过没有用身体近距离地去碰铁棒。此后，还有一些研究者也做了类似的实验，在俄罗斯有一位物理学家模仿这个实验时，因为操作不慎被雷电击死



几年后，有传闻和一些书籍称富兰克林用风筝替代了铁棒，做了这个著名风筝雷电的实验，但缺乏充分的证据。后来有研究者发现富兰克林本人从没有正式承认做过这个实验。尽管对于富兰克林是否做过风筝实验存在争议，但有一点可以肯定的是，富兰克林即使做过风筝实验，也肯定不会和传说中的一模一样。

### 实验若成功，富兰克林必死

如果云层上的电荷聚集越来越多，和地面之间形成的电压越来越大，最后它们击穿十几千米厚的空气，形成一条到达地面的导电通道，释放巨大的能量，这就产生了雷电。如果这条通道正好途经某人的身体，放电电流会很大，数量级达到几十千安甚至几百千安以上，那么大的电流流过人的躯体，首先伤害的是受害者的大脑和心脏。因为几毫安的电流就可以使人类的心脏发生心室纤维性颤动、停止搏动。雷电流也会致使呼吸系统麻痹而停止呼吸，从而使人丧命。此外，雷电流的极大的机械效应足以撕裂皮肤和肌肉，而强烈的热效应也足以烧焦躯体。这种雷击事故称为“直接雷击”。遭受直接雷击的人十有八九会死亡，即使没有死亡也会重伤。如果这条导电通道没有直接通过人体，相隔一段距离，比如击中附近的一棵树，人体仍然有可能因为感应的电流而触电，称为“感应雷击”，感应雷击有时会比较弱，被击中者无大碍，受雷击大难不死的幸运儿大多数是这种情况

如果按故事中的情节，风筝被雷电击中，雷电电流顺着风筝线一直

到钥匙，富兰克林的手指与钥匙之间的距离越近，而且之间产生了明亮的火花，富兰克林这样直接被雷击中，绝不可能安然无恙

### 实验！实验！

为了查明真相，著名实验帝《流言终结者》在第4季第5集里复制了这个流言。他们试图证明三件事：

- (1) 风筝能否吸引电流，并且通过长长的风筝线传递到钥匙；
- (2) 注入钥匙的电流量是否足以电到富兰克林的手指；
- (3) 那股电流是否足以让放风筝者心跳停止

实验者模仿18世纪时使用的材料制作了一个大风筝和木板棚架，把风筝在天气晴朗的海滩上放飞，在海风中，虽然完全没有电闪雷鸣，但空气中的电荷和风筝线与空气之间的摩擦产生的电荷已经可以使风筝明显地带上静电，风筝线上挂着的钥匙在吱吱响，第一件事很容易地验证了

接着他们把风筝弄湿，风筝上的静电量进一步增大，但把手指靠近钥匙，却没有出现明显的触电感觉。为了增大电量，终结者找了一个大金属球形状的电荷产生器代替海边空气作为电荷来源，这个大金属球产生的电荷远远高于空气中的静电产生的电荷，但比起真正的闪电还是微不足道的。当风筝靠近这个金属球时，就会被击中，如果把一个探头靠近钥匙，可以看到两者之间有微弱的火花。第二件事也证实了

为了模拟雷电的威力，终结者走进了电力公司的试验中心，这里的高压电可达到100万伏，可是比起真正的雷电1亿伏的电压，也只有1%，他们用组织替代胶制作了一个假人模型，里面安装了一个模拟的心跳检测器，并用模拟的雨水淋湿的风筝线进行实验。风筝被高压电击中时，在钥匙和假人的“手指”之间出现了明亮的电弧，通过模拟心脏的电流已经超过了可以使人心脏停跳的最大电流的很多倍。这个“迷你版本”已经中心让富兰克林英勇牺牲很多次了。由此可见，富兰克林在直接被雷电击中后还毫发无损，并且淡定地说“我可以证明闪电是电！”是不靠谱的

### 结论：谣言粉碎

富兰克林或许有过风筝实验的想法，即便他真的做过这个实验，也只可能是被风筝上带的一些静电电到。如果被真的雷电击中的话，就不会是手被电麻了，很可能是当场暴毙

到现在为止，我们已经见识了很多类型的电源，通常衡量一个电源的重要指标是它的电压，也就是它产生电流的能力。不同的电源，所提供的电压也不相同，世界上第一个电池是由意大利人伏特于1800年发明的，后来物理界就用他的名字作为衡量电压大小的单位，简称“伏”，或者用大写字母V表示

我们平时所使用的电池电压是5V，如果电压超过36V（这个电压称为安全电压，如果电压低于这个电压，则不会危害到人），会觉得手臂发麻，有电击感；

## 1.5 电路和电路图

电灯泡的发明用到了电子在导体中流动时的一个特点，电子在导体中流动时并不是那么顺畅，差不多是在原子的密林中磕磕碰碰，换句话说，导体实际上对它的流动具有阻碍作用，在电学中，影响电子流动的这种特性叫电阻

电阻的作用是让电子的运动不那么顺畅，它不是导体独有的，通常情况下所有的物体都有电阻，一些电阻较小的物质称为导体；而另一些电阻很大很大的物质被称为绝缘体。在正常情况下，同样长的一段电线，用银来制作的话，电阻是最小的（遗憾的是，银又是最昂贵的物质之一）

从能量的角度来看，电子的流动是因为电源赋予了电子能量。“能量”这种说法我们差不多每个人都听说过，但通常也只是停留在意会的层面上，因为谁也看不到它，更无法触摸。但能量却无处不在。

关于能量的一个很重要，同时也是很有意思的特点是，它可以从一个物体传递到另一个物体，也可以从一种形式转化成另一种截然不同的形式。烈日当空，被太阳晒到的地方就会发热，这说明太阳能转变成了热能。风吹在风力发电机上，能发电，风能转变成了电能。太阳能也可以转变成化学能贮存在我们吃的植物里，比如大米、水果和蔬菜。

总之，用物理学上的说法就是“能量即不会凭空产生，也不会凭空消失，它只能从一种形式转化为别的形式，或从一个物体转移到别的物体，在转化或转移的过程中其总量不变。”这叫做能量守恒定律，1847年由德国物理学家、生理学家赫尔姆霍茨（1821-1894）首先提出。



# 能量守恒定律

能量守恒定律是自然科学中最基本的定律之一，可表述为“在孤立系统中，能量从一种形式转换成另一种形式，从一个物体传递到另一个物体，在转换和传递的过程中，各种形式，各个物体的能量的总和保持不变。整个自然界也可看成一个孤立系统，而表达为自然界中能量可不断转换和无反应性，但总量保持不变。

能量是无处不在的，当电源开始工作的时候，它所自己的能量源源不断地传递给电子，电子携带着能量，通常也是在能量的作用下开始流动形成电流

由于电阻的存在，电子会在电线中释放出一部分能量，使得组成电线的原子比平时格外活跃 – 电线发热了。通常情况下这不是个太大的问题，充其量只是一部分电能白白浪费掉了，而电线上的热量也会很快散发到空气中。但在另外一些极端的情况下，由于电压太高，电流太大，产生的热量不能及时发散，就会烧红电线，导致火灾。从另一个角度来看，这也就是灯泡能够发光的工作原理

一个灯泡、一段电线、一节电池这几样东西装配在一起就能工作得很好，这称为电路（直观上理解的话就是“电流或电子的通路”）。组成一个电路有几样东西是必需的：电源、导线和依靠电工作的各种器件。如果不通过用电器直接用电线接通电源的两极，这叫做短路。电子像潮水一样涌动的时候当然是携带了能量的，这些能量要么传递给用电器，要么将电源烧坏，电线烧红 – 总之，当它们被逼着从电源里怒气冲冲地跑出来的时候，当然需要一个撒气的地方，除非赶快想办法将电路断开

不同的电路有不同的用途，而为了不同的用途也需要发明不同的电路。采用一些简单的符号来代表各种电路器件。一些比较简单的，如电池、灯泡和电线，它们的实物图及符号如图1.3所示。



图1.3 电池、灯泡和电线的实物图及符号

有了这些符号后，再来画图1.1的电路就很省事儿了（图1.4）

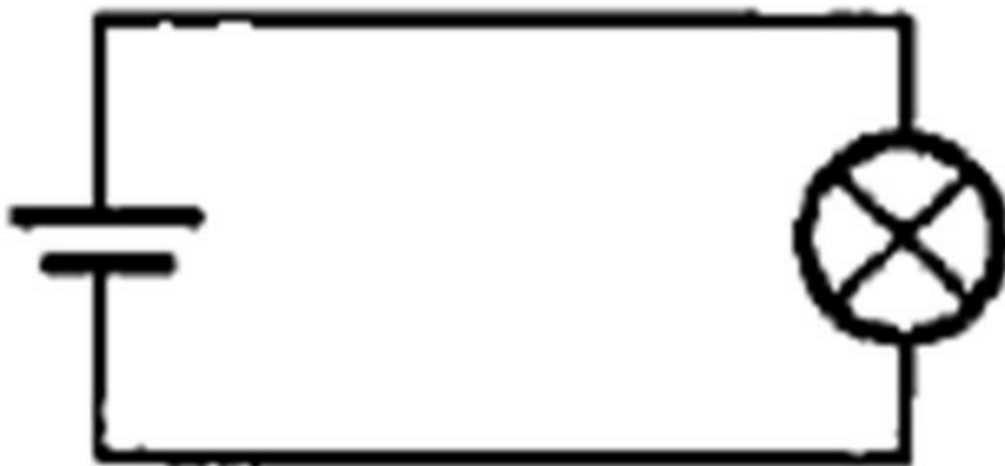


图1.4 电路的符号化表示

电路复杂的时候，会经常遇到电线交叉的情况。如果它们并没有连通，就表示成图1.5(a)那样；如果它们是连接在一起的，则表示成图1.5(b)那样

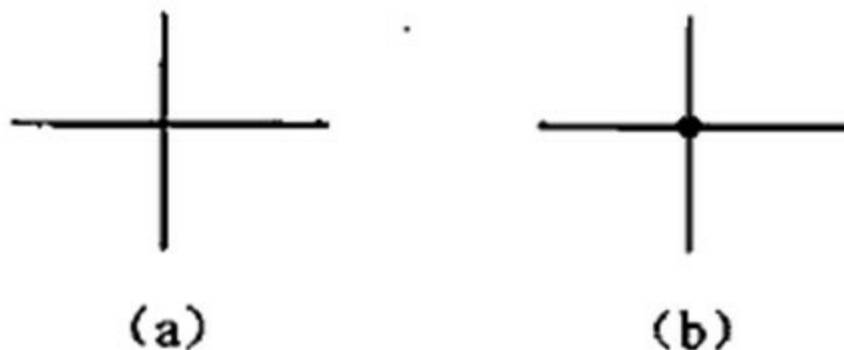


图1.5 线路交叉的两种情况

这里有一个很好的例子说明了电线是如何交叉连接的，如图1.6所示。它表明了如何用一节电池让两只灯泡同时发光，右边是实物图，左边是它的电路图



图1.6 一个电路图的例子

另外，能够随意控制电流的通断是很重要的，这样需要另外一样东西——开关。为了表示一只开关，通常使用下面这样的符号（图1.7）



图1.7 开关的符号

而要用一只开关来控制两个灯泡的亮灭，它的电路图则应当是如图1.8所示的那样

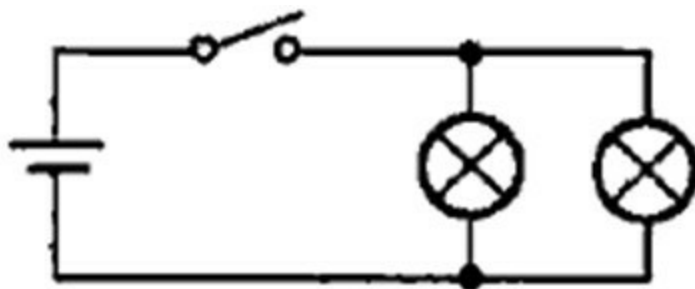


图1.8 用开关组成的电路

# 补充

公元前600年左右，希腊人泰勒斯（Thales）就发现摩擦过的琥珀能吸引轻小物体的现象。16世纪英国御医威廉姆·吉尔伯特（William Gilbert）在研究这类现象时首先根据希腊语“琥珀”创造了英语中的electricity（电）这个词，用来表示琥珀经过摩擦以后具有的性质，并且认为摩擦过的琥珀带有电荷（**electric charge**）。后来人们发现很多物质都会由于摩擦而带电，并且带电物体之间存在着相互排斥或相互吸引的作用。

摩擦后的物体所带的电荷有两种：用丝绸摩擦过的玻璃棒所带的电荷是一种；用毛皮摩擦过的硬橡胶棒所带的电荷是另一种。同种电荷相斥，异种电荷相吸。没有发现对上述两种电荷都排斥或都吸引的电荷，这表明，自然界的电荷只有两种。富兰克林（Benjamin Franklin）把前者命名为正电荷（**positive charge**），把后者命名为负电荷（**negative charge**）。

构成物质的原子本身就是由带电粒子构成的，带正电的质子和不带电的中子构成原子核，核外有带负电的电子。原子核的正电荷数量与电子的负电荷数量一样多，所以整个原子对外界较远位置表现为电中性。

原子核内部的质子和中子被核力紧密地束缚在一起，核力来源于强相互作用，所以原子核的结构一般是很稳定的，核外电子靠与质子的吸引力维系在原子核附近，通常离原子核较远的电子受到的束缚图小，容易受到外界的作用而脱离原子，当两个物体互相摩擦时，一些束缚得不紧的电子从一个物体转移到另一个物体，于是原来电中性的物体由于得到电子而带负电，失去电子的物体则带正电。这就是摩擦起电

（**electrification by friction**）的原因。

不同物质的微观结构不同，核外电子的多少和运动状况也就不同，而且由大量原子或分子组成物质时，由于原子或分子间的相互作用，核外电子的运动状况也会有所变化。这些情况都使不同物质中的电子在受到外界作用时产生不同结果。例如，金属中离原子核最远的电子往往会脱离原子核的束缚而在金属中自由活动，这种电子叫做自由电子（**free electron**），失去这种电子的原子成为带正电的离子（**ion**），它们在金属内部排列起来，每个正离子都在自己的平衡位置上振动而不移动，只有自由电子穿梭其中，这就使金属成为导体。

当一个带电体靠近导体时，由于电荷间相互吸引或排斥，导体中的自由电子便会趋向或远离带电体，使导体靠近带电体的一端带异号电

荷，远离的一端带同号电荷，这种现象叫做静电感应（**electrostatic induction**），利用静电感应使金属导体带电的过程叫做感应起电（**electrification by induction**）

## 电荷守恒定律

无论是摩擦起电还是感应起电，本质上都是使微观带电粒子（如电子）在物体间或物体内部转移，而不是创造了电荷。

电荷即不能创造，也不能消灭，只能从一个物体转移到另一个物体，或者从物体的一部分转移到另一部分，在转移过程中，电荷的总量保持不变。这个结论叫做电荷守恒定律（**law of conservation of electric charge**）

近代物理实验发现，在一定条件下，电荷是可以产生和湮没的，例如，由一个高能光子可以产生一个正电子和一个负电子；一对正、负电子可同时湮没转化为光子。在这种情况下，带电粒子总是成对产生或湮没，两个粒子带电数量相等但正负相反，而光子又不带电，所以电荷的代数和仍然不变，因此，电荷守恒定律也常表述为：一个与外界没有电荷交换的系统，电荷的代数和总是保持不变的。这是自然界重要的基本规律这之一

### 元电荷

电荷的多少叫电荷量（**electric quantity**），在国际单位制中，它的单位是库仑（**coulomb**），简称库，用C表示，正电荷的电荷量为正值，负电荷的电荷量为负值。

迄今为止，科学实验发现的最小电荷量就是电子所带的电荷量。质子、正电子所带的电荷量与它相同，但符号相反，人们把这个最小的电荷量叫做元电荷（**elementary charge**），用e表示，实验表明，所有带电体的电荷量或者等于e或者是e的整数倍。这就是说，电荷量不能连续变化的物理量

电荷量e的数值最早由密立根（Robert Andrews Millikan）测得的。在密立根实验后，人们又做了许多测量，现在测得的元电荷值为

$$e=1.60217733\times 10^{-19}\text{C}$$

## 库仑定律（**Coulomb law**）

真空中两个静止点电荷之间的相互作用力，与它们的电荷量的乘积成正比，与它们的距离的二次方成反比，作用力的方向在它们的连线上

电荷间这种相互作用力称为静电力（**electrostatic force**）或库仑力

什么是点电荷？任何带电体都有形状和大小，其上的电荷也不会集中在一点上，当带电体间的距离比它们自身的大小在得多，以至带电体

的形状、大小及电荷分布状况对它们之间相互作用力的影响可以忽略不计，这样的带电体就可以看作带电的点，叫做点电荷（**point charge**）。可见点电荷类似于力学中的质点，也是一种理想化的物理模型

$$F = k \frac{q_1 q_2}{r^2}$$

k是比例系数，叫做静电力常量（**electrostatic force constant**）

在国际单位制中，电荷量的单位是库仑（C），力的单位是牛顿

$$F = k \frac{q_1 q_2}{r^2}$$

（N），距离的单位是米（m），所以 中各物理量的单位都已确定，k的数值由实验测定，结果是

$$k=9.0 \times 10^9 \text{N} \cdot \text{m}^2 / \text{C}^2$$

这就是说电荷量为1C的点电荷在真空中相距1m时，相互作用力是  $9.0 \times 10^9 \text{N}$ ，差不多相当于一百万吨的物体所受的重力！由此可见，库仑（C）是一个非常大的电荷量单位，天空中发生闪电前，巨大的云层中积累的电荷可达几百库仑。

## 电场强度

### 电场

万有引力曾被认为是一种既不需要媒介，也不需经历时间，而是超越空间直接发生的作用力，并被称为超距作用。库仑定律似乎表明，静电力像万有引力一样，也是一种超距力

19世纪30年代，法拉第（**Michael Faraday**）提出一种观点，认为在电荷周围存在着由它产生的电场（**electric field**），处在电场中的其他电荷受到的作用力就是这个电场给予的。

近代物理学的理论和实验证实并发展了法拉第的观点，电场及磁场已被证明是一种客观存在，并且是互相联系的，统称为电磁场

（**electromagnetic field**）。变化的电磁场以有限的速度 – 光速在空间传播。电磁场和分子、原子组成的实物一样具有能量、质量和动量，因而场与实物是物质存在的两种不同形式。

只有在研究运动的电荷，特别是运动状态迅速变化的电荷时，上述电磁场的实在性才突显出来。静电荷产生的电场，称为静电场

（**electrostatic field**）

### 电场强度

电场明显的特征之一是对场中其他电荷具有作用力。但不能直接用试探电荷所受的静电力来表示电场的强弱，因为对于不同的电荷q，即

便在电场中的同一点，所受的静电力 $F$ 也不相同，但实验表明，在电场

中的同一点，比值 $\frac{F}{q}$ 是恒定的，在电场中的不同的点，比值 $\frac{F}{q}$ 一般是不同的。这个比值由电荷 $q$ 在电场中的位置决定，与电荷 $q$ 的电荷量大小无

关，这个比值才是反映电场性质的物理量，在物理学中，就用比值 $\frac{F}{q}$ 来表示电场的强弱

放入电场中某点的电荷所受的静电力 $F$ 跟它的电荷量 $q$ 的比值，叫做该点的电场强度（**electric field strength**）。用 $E$ 表示电场强度，则有：

$$E = \frac{F}{q}$$

电场强度的单位应是牛[顿]每库[仑]，符号为N/C

### 电场线

形象地了解和描述电场中各点电场强度的大小和方向很重要，法拉第采用了一个简洁的方法描述电场，那就是画电场线（**electric field line**）

电场线是画在电场中的一条条有方向的曲线，曲线上每点的切线方向表示该点的电场强度方向。电场线不是实际存在的线，而为了形象地描述电场而假想的线。电场线有以下几个特点：

1 电场线从正电荷或无限远出发，终止于无限远或负电荷

1 电场线在电场中不相交，这是因为在电场中任意一点的电场强度不可能有两个方向

1 在同一幅画中，电场强度较大的地方电场线较密，电场强度小的地方电场线较疏，因此可以用电场线的疏密来表示电场强度的相对大小

## 电势能和电势

建立了电场强度的概念，知道它是描述电场性质的物理量。倘若把一个静止的试探电荷放入电场中，它将在静电力的作用下做加速运动，经过一段时间后获得一定的速度，试探电荷的动能增加了，这是电力做功的结果，功又是能量变化的量度。那么，在这一过程中，是什么能转化成试探电荷的动能？为此，首先要研究静电力做功的特点

电力做功与电荷的起始位置和终止位置有关，与电荷经过的路径无关。

电荷在电场中也具有势能，这种势能叫做电势能（**electric potential energy**或**electrostatic potential energy**）

物体在地面附近下降时，重力对物体做正功，物体的重力势能减少；物体上升时，重力对物体做负功，物体的重力势能增加。与此相似，当正电荷顺电力线移动时，静电力做正功，电荷的电势能减少；当正电荷逆电力线移动时，电荷克服静电力做功，电荷的电势能增加。

功是能量变化的量度，所以：静电力做的功等于电势能的减少量。静电力做的功只能决定电势能的变化量，而不能决定电荷在电场中某点的电势能的数值。只有先把电场中某点的电势能规定为零，才能确定电荷在电场中其他点的电势能。电荷在某点的电势能等于静电力把它从该点移动到零势能位置时所做的功

通常把电荷在离场源电荷无限远处的电势能规定为零，或把电荷在大地表面上的电势能规定为零

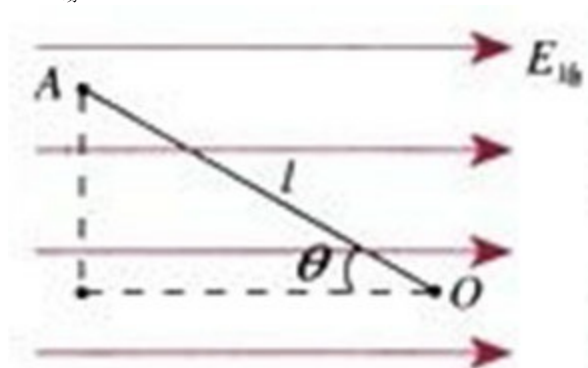
重力或引力存在的空间也称为重力场或引力场，物体在重力场或引力场中移动时，重力或引力做的功，跟电荷在电场中移动时静电力做的功虽然相似，但还是有很大的差异。这是由于存在两种电荷的缘故。在同一电场中，同样是顺电力线，移动正电荷与移动负电荷，电荷的电势能的变化是相反的。

## 电势

电势也是表征电场性质的重要物理量

从电荷在电场中的电势能与它的电荷量的比值来研究电势。

有一个电场强度为 $E_{\text{场}}$ 的匀强电场，如下图：



规定电荷在O点处的电势能为零，A为电场中任意一点，电荷 $q$ 在A点的电势能 $E_{\text{pA}}$ 等于电荷 $q$ 由A移至O点的过程中静电力做的功。由于静电力做功与路径无关，为方便起见，选择直线路径AO进行计算。设AO的长度为 $l$ ，则 $E_{\text{pA}} = qE_{\text{场}}l\cos\theta$ 。可见，电荷 $q$ 在任意一点A的电势能 $E_{\text{pA}}$ 与 $q$ 成正比，也就是说，处于A点的电荷，无论电荷量大小是多少，它的电势能

$$\frac{E_{\text{pA}}}{q}$$

与电荷量的比值  $q$  都是相同的。对电场中的不同位置，由于 $l$ 和 $\theta$ 可以不同，所以这一比值一般是不同的。



从以上分析可知，电荷的电场中某一点的电势能与它的电荷量的比值是由电场中这点的位置决定的

电荷在电场中某一点的电势能与它的电荷量的比值，叫做这一点的电势（**electric potential**）

国际单位中，电势的单位是伏特（**volt**），符号为V。在电场中的某一种，如果电荷量为1C的电荷在该点的电势能是1J，这一点的电势就是1V，即 $1V=1J/C$

电场线指向电势能降低的方向

与电势能的情况相似，应该先规定电场中某处的电势为零，然后才能确定电场中其他各点的电势。在物理学的理论研究中取离场源无限远处的电势为零，在实际应用中常取大地的电势为零

电势只有大小没有方向，是标量

### 电势差

用不同的位置作为测量高度的起点，同一地方的高度的数值就不同，但两个地方的高度差却保持不变，同样的道理，选择不同的位置作为电势零点，电场中某点电势的数值也会改变，但电场中某两点间的电势的差值却保持不变，正因为这个缘故，在物理学中，有时电势的差值比电势更重要

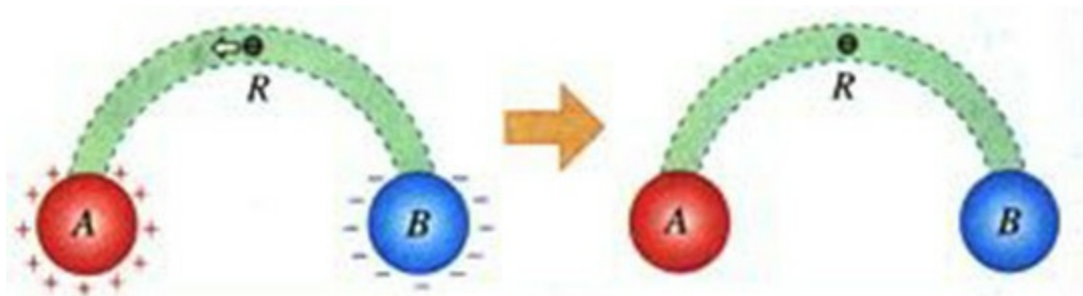
电场中两点间电势的差值称为电势差（**electric potential difference**），也叫电压（**voltage**）。

### 导体中的电场和电流

闪电时强大的电流使天空发出耀眼的光，但它只有一瞬间，而手电筒的小灯泡却能持续发光，这是为什么？

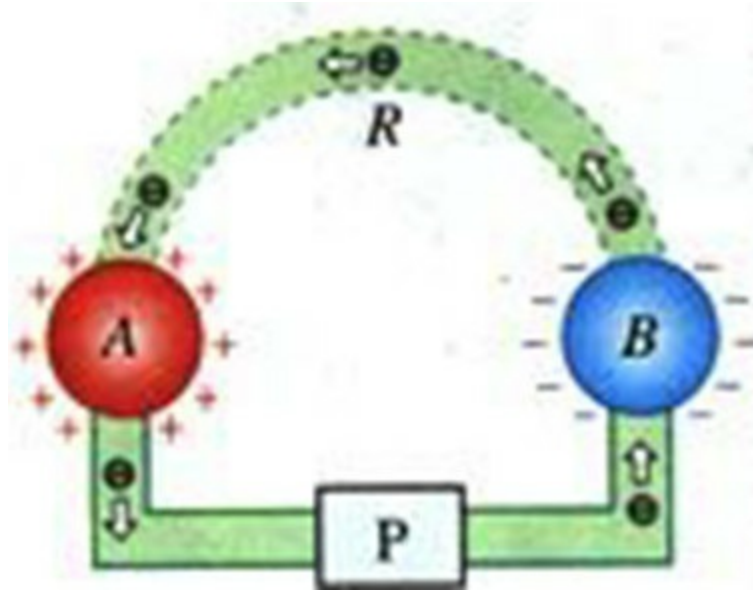
### 电源

有A,B两个导体，分别带正负电荷，它们的周围存在着电场，如果在它们之间连接一条导线R，如下图：



导线R中的自由电子便会在静电力的作用下定向运动，B失去电子，A得到电子，周围电场迅速减弱，A，B之间的电势差很快消失，两导体成为一个等势体，达到静电平衡，在这种情况下，导线R中只可能存在一个瞬时电流

倘若在A,B之间连接一个装置P（如下图）



此装置P能源源不断地把经过导线R流到A的电子取走，补充给B，使A，B始终保持一定数量的正、负电荷，这样A,B周围空间（包括导线中）始终存在一定的电场，A,B之间便维持一定的电势差，由于这个电势差，导线中的自由电子就能不断地在静电力作用下由B经过A定向移到，使电路中保持持续的电流。上图中能把电子从A搬运到B的装置P就是电源（**power source**）

在金属导体中，能够自由移动的是自由电子

在外电路中，自由电子从负极流向正极，电源之所以能维持外部电路中稳定的电流，是因为电源有能力把来到正极的自由电子经过电源内部不断地搬运到负极。

由于正、负极总保持一定的正、负电荷，所以电源内部总存在着由正极指向负极的电场，在这个电场中，自由电子所受的静电力阻碍它向负极移动，因此在电源内要使自由电子向负极移动就一定要有“非静电力”作用于自由电子才行，也就是说，电源把自由电子从正极搬运到负极的过程，这种非静电力在做功，使电荷的电势能增加。

在电池中，非静电力是化学作用，它使化学能转化为电势能；在发电机中，非静电力是电磁作用，它使机械能转化为电势能。所以，从能量转化的过程来看，电源是通过非静电力做功把其他形式的能转化为电势能的装置

电源移动电荷，增加电荷的电势能，与抽水机抽水增加水的重力势能很相似。

电源内部也是由导体组成的，所以也有电阻，这个电阻叫电源的内阻（**internal resistance**）

## 第2章 用电来表示数

计算机这三个字只是一个笼统的概念，泛指一切具有计算功能的机器。这样说来，计算机就多了，比如我国的算盘。后来人们又发明了各种各样的机械计算机。

20世纪之后，电学开始大发展，电子计算机就这样出现了。电子计算机也是计算机的一种，而且是最成功、应用最广泛的一种，以至于提起计算机，人们都会想到电子计算机。

电子计算机俗称“电脑”，但好像只有在我们国家才这样说，原因可能是大家觉得它和大脑一样擅长计算，甚至在某些方面比大脑的工作更有效。最早的时候，人们发明计算机的目的仅仅是用来进行数学计算，即使是几十年前，当世界上第一台电子计算机出现的时候，研制它的目的依然是进行数学计算，这一点没有改变。说到这里，大家可能觉得这与现实情况不同，现代计算机功能太多了，既能上网又可以写文章排版打印、听音乐、看电影、玩游戏.....，但所有这一切看不出与数学运算有什么必然联系

这种看法并不正确。在任何一台现代的计算机内部，数学运算仍是最重要的组成部分之一，而且是非常基础的组成部分

## 2.1 怎样用电来代表一个数字

要进行数学计算，首先要解决的问题是如何将参与计算的数送进计算机。在机械计算机的时代，人们一般是通过一些精心设计的零件移动到合适的位置来做到这一点。但对于现代电子计算机来说，情况则完全不同。它不是电动的 – 就像用电动机代替手摇脚踏，或者用电动机代替驴来推磨那样。相反，它从里到外都是电气化的，用电来表示数字，用电进行计算。

通常，数学运算被构造成一个独立的部件，这个部件就像一个盒子，它从外面接收一些数，经过计算后，再把结果送出来。

制造一个包括所有数学运算功能的部件固然好，但对刚看完第1章的你来说显然不切合实际。最明智的做法是先制作一个小的、能完成某个简单运算的部件。当这个部件制作完成后，根据需要再进行扩充，看起来加法运算非常简单，那我们就从制造一个加法运算部件开始吧

鉴于所有的电器都被放在一个盒子里，所以一个加法运算部件看起来就像这样（图2.1）

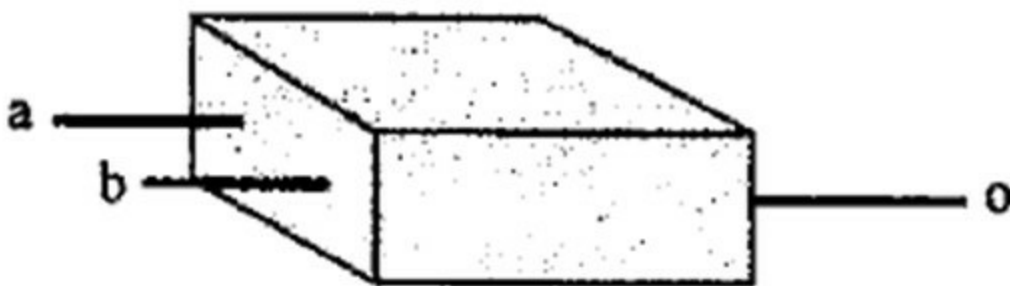


图2.1 加法运算部件

因为加法运算需要一个加数和一个被加数，所以这个加法运算部件提供了a,b两个输入端，好让它知道要算的数是什么。当这个加法运算部件完成计算后，把结果从o端送出来。由于刚刚学习了电学知识，所以现在到了发挥想象力的时候了：你认为应该怎样通过a和b将数据送到这个部件里？

要想把准备加起来的两个数通过a和b送到运算部件里，最自然的想法就是将不同的数表示成不同的电压。

这个想法太奇妙了！不是吗？如果我要计算 $20+15$ ，我可以在a端加20V电压而在b端加15V电压，当运算完成后，o端输出35V电压 – 这正是我们所要的结果

遗憾的是这种美好的愿望会因为一个无奈的事实而注定无法成为现实。好的设计要在灵光一闪后要反复推敲。在上面的设计中，当参与运

算的数都很小时，它当然可以工作得很好，但当数字变得很大时（这是最常见的情况），情况开始变得微妙，比如计算 $99768332+112211$ ，这意味着得产生9000多万伏的高压

并不说人类无法得到这样高的电压，事实上这很容易，但这个运算部件未必能够承受住这样的高压而不被烧毁。就算我们真能制造出这样的机器，恐怕谁也不敢靠近它，更不要说把它买回去放在家里。这样的计算机最好还是放在一般人到不了的地方，并在醒目的位置贴上“内有高压，请勿靠近”的标签

即使这样可行，那么制造这样一台运算部件真正无法逾越的困难是表示像11.00156这样的小数，通常，一个电路只能工作在近似精确的状态，因为有很多不可预知的因素会产生干扰。除了电路本身要消耗电能外，像温度变化、组成电路的零件出故障这样的情况也会偶尔出现，这些都能导致整个电路的状态产生一些微小的改变。这意味着当你计算 $20+15$ 的时候，尽管从a,b送进去的是精确的20V和15V电压，从o端输出可能不正好是35V，可能高或低，取决于具体的情况，但总的说来这并不算是个什么大事儿，因为我们知道自己计算的是整数，尽管不太准确，高一点低一点这个结果是我们接受的

不过麻烦在于，假如我们真的想得到一个精确的结果11.00156时，该怎么办呢？将电压精确地调整到这个数值是非常麻烦的，而最要命的是这个运算部件根本无法保证它不会变化。如果正在进行的金融计算，这个误差就离谱了，总之得换一个考虑问题的思路

经过一段时间的思考，你可能会想到另一个办法来解决这个办法。前面的方案之所以行不通，是因为仅仅只用一根导线是不可能表示所有的数的，同时发现，无论一个数有多大，它问题0，1，2，3，4，5，6，7，8，9的不同组合。比如125是1，2，5的组合；93850是9，3，8，5，0的组合等等。有了这个发现后，我们不再使用单独的一根导线，而是使用多根导线来表示一个数，其中每根导线都对应着这个数中的一位，如图2.2所示

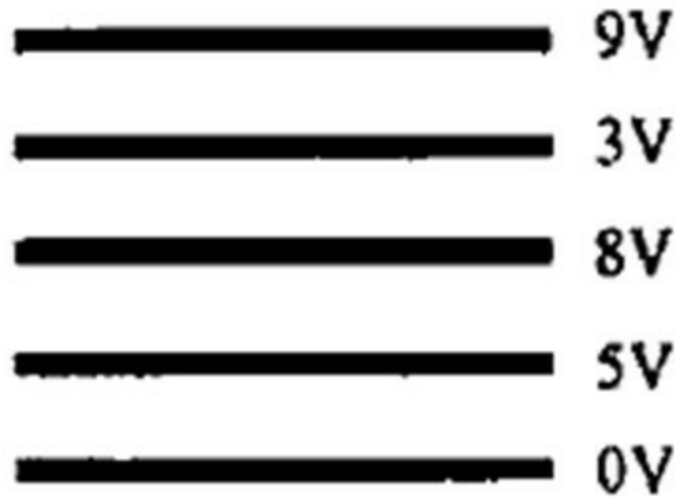


图2.2 用多根导线来表示一个数

这个修改是成功的，在图2.2中，5根导线中的每一根分别代表着93580这个数的一位，按从上到下的顺序，在具体应用的时候，根据这个数每一位的数值为各个导线分配相应的电压，它最大的特点就是不再使用令人畏惧的高电压，取而代之的是从0-9V的九种低电压，如果觉得以伏为单位还是太高，有点浪费的话，你也可以使用更小的电压，比如毫伏（mV）来代替，完全不影响效果。

在这个例子中，数据输入端a,b已经被分别扩充为5根导线，当然用这5根导线可以表示的数最大为99999，并不算大，如果需要可以使用更多的导线，使用尽可能多的导线是必要的，比如下面这个加法运算部件，它是在图2.1的基础上修改而来的（图2.3）



图2.3 修改后的加法运算部件

到目前为止一切还好，遗憾的是没有说明它如何表示一个小数，要表示一个小数，有多种办法可供选择。最简单、最省事儿的办法就是把导线分成两组，分别代表整数部分和小数部分

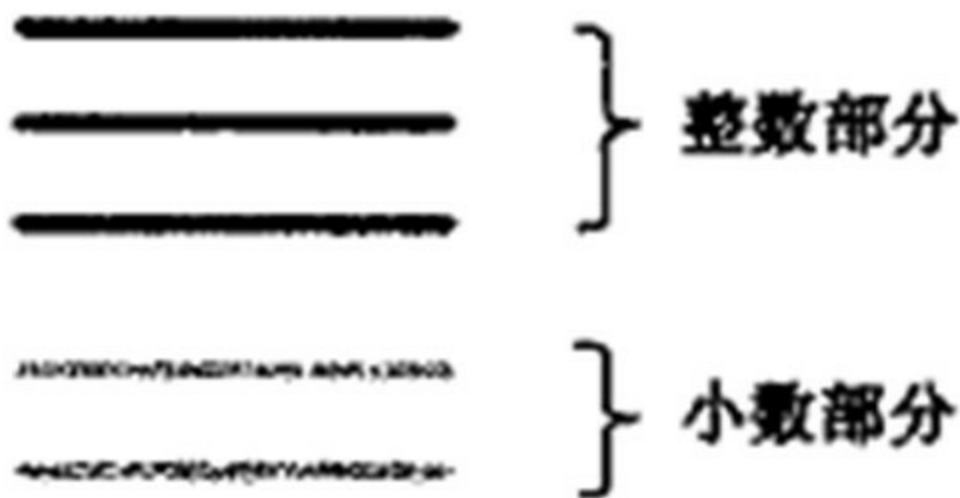


图2.4 整数部分和小数部分的划分

划分的方法可以随意，如上图那样，把5根导线划分成3位整数部分和2位小数部分。如果用9根导线，就可以划分7位整数部分和2位小数部分或6位整数部分和3位小数部分等等

这个方案能保证数据的精确度吗？完全可以。不管一个数有大，也不管它是不是有小数部分，只要每一位对应一根导线，将一个数的每一位分开，这是保证一个数在传输和处理过程中不会发生变化的第一个重要举措

第二条措施也同样重要，由于每一位可以是0-9中的任何一个数字，所以它意味着应当为每根导线准备10种电压（比如从0V-9V）。当然，电压可能不会十分精确，可以规定如果电压在6.5-7.4V之间，认同等于7V；电压在7.5-8.4V之间认同等于8V，这样就很好地解决了精确度的问题，除非发生比较大的电路故障

这个用电来表示数并进行简单加法运算的方案无论从哪个方面看都无懈可击，所以下一步是如何具体实现这个运算部件。

这个方案理论上来说是可选的，模拟计算机就是按这种思路做成的，模拟计算机在1940年以前就有了，甚至被安装在潜艇上，用来计算发射鱼雷所需的方向和速度。但模拟计算机实现起来很困难，完全可以采用更好的方法，而且只用很普通材料就可手工实现

## 2.2 古怪的二进制计数法

这一节讨论如何数数。比如，图2.5中一共有几棵树？肯定是12棵



图2.5 上面这些树的个数可记做“12”

为什么要把这些树的个数记做“12”而不是其他的符号呢？这是因为我们平时使用的这种记数方法叫做十进制记数法，十进制记数法只有十个符号：0，1，2，3，4，5，6，7，8，9，可以表示任意的数，奥秘在于组合。

十进制有十个符号，9是最大的，当要表示更大的数，将9变成0，然后向左进一位，记做10，如图2.6所示，当满10的时候向前进位，这就是十进制记数法的由来

**9**  
把9变成0  
**1 0**  
往左进一

图2.6 十进制是逢“十”进位的

那么二进制记数法是什么呢？在二进制记数法中，只有0，1两个符号，那么1+1怎么解决呢？用“2”表示，行不通，因为二进制记数法只有



0和1，没有更多的符号，所以也是将1变成0，向左进位来表示的  
尽管二进制记数法只有两个符号，也可以有无穷无尽的组合。

## 2.3 二进制数就是比特串

十进制数有不同的数位，个位、十位、百位.....，但在二进制里通常不需要这样细致的划分，因为二进制数一般都很长，对于单个二进制数位，只有一个称为“比特”，每个比特具有两个可能的值：0或1

最初二进制中的每一位在英语里被表示为Binary digit，意思是“二进制数位”或“二进制数字”。约翰·怀尔德·图基（John Wilder Tukey，美国著名数学家）想用一种更短小的名称以方便交谈和书写，他一开始想到的是bigit和binit，但最终他选择使用bit这个单词，缩写为b，当这个词传入中国时，音译为“比特”

## 2.4 用开关表示二进制数字

学会了二进制，运算部件制作计划又开始缓慢向前推进。当然，在这个过程中首要问题还是解决如何方便地用电来表示具体的数

二进制只有0和1两个符号，这可以用开关来实现：

开关断开时，电流被切断，代表0；

开关接通时，电路中有电流通过，代表1，如图2.16所示



图2.16 开关的断合对应着0和1

另一种可行的方案与此相反，开关断开表示1，开关接通表示0，但对大多数人来说，有些别扭，毕竟我们习惯了把“没有”看成0，所以我们不使用这种方案，虽然它也是可行的。

因为在大多数情况下一个真正的二进制数不仅只是一个0或一个1，它可能包含了很多比特，是一连串的0，1，所以要表示一个真正的二进制数，比如 101（也就是十进制数的5），就需要一排开关，每一个开关对应一个比特（图2.17）

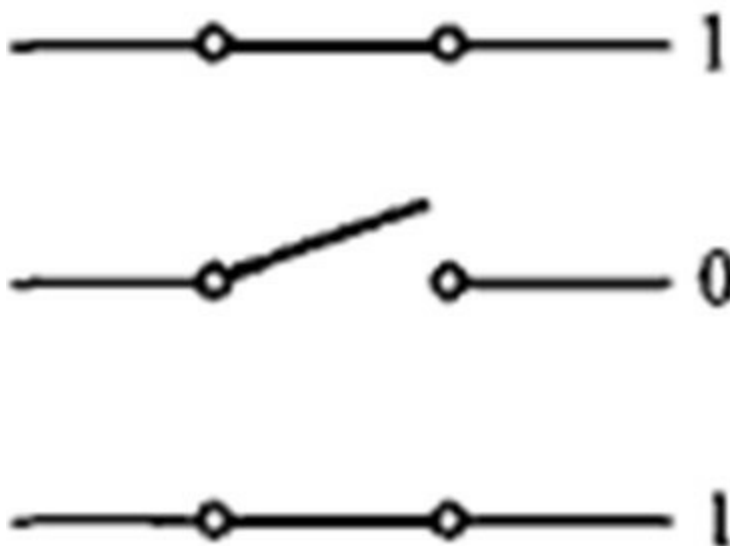


图2.17 通过使用多个开关可代表任何二进制数

这个创意很新颖，也非常重要，所以我们应该立即将它应用到我们正在努力制造的运算部件中，如图2.18所示

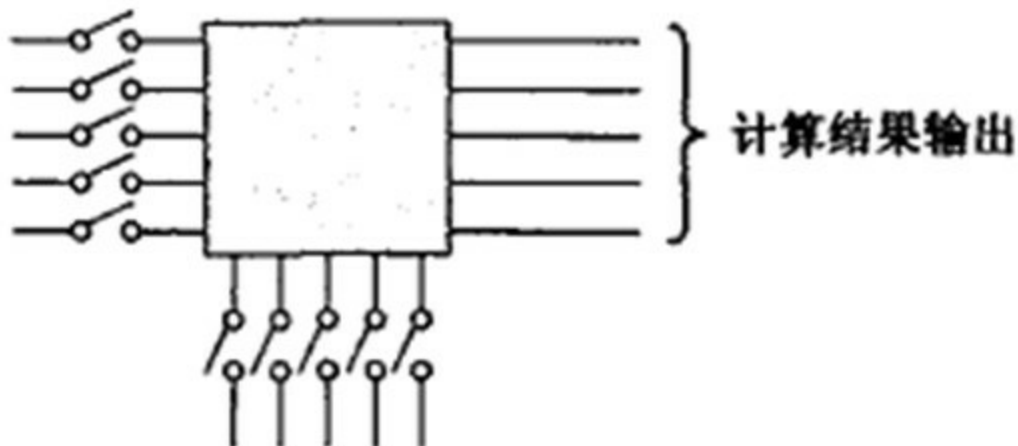


图2.18 理想中的二进制运算部件

图中的方框通常代表一个具有某种功能的电路，在这里它代表的是我们一直努力要制造的运算部件，之所以这么做，是因为我们正在讨论的是如何用开关来表示数，这是我们当前关注的焦点，况且，现在还不知道它的内部到底如何构造

这个运算部件的左边和下面各有5个开关，分别用于输入两个参与运算的二进制数，这意味着它们都是5比特的。要表示一个二进制数，只需要接通或断开它们中的一个或多个，那么为什么这里非得是两个5比特二进制数呢？没有什么特别的原因，这只是一个例子，你可以使它只有2比特或者20比特，比特越多，需要的电线和开关就越多，相应的运算部件能计算的数就越大

这个图充分表明了二进制数之所以在电的世界里受到欢迎的原因，要在以前，必须制作一大堆电路，为的是生成不同的电压，这还不算，为了知道生成的电压是否符合要求，还得拿着电压表一遍一遍地逐个测量。现在只需要准备一个合适的电源和为数不多的开关就足够了。至于精度，在这里有电表示1，没有电表示0，使用多大的电压都无所谓，只要不会烧坏，你认为这里精度会是问题吗？

除此之外，还有更令人感到振奋的，在前面的设计过程中，由于忙着解决如何将数输送到运算部件里去，还没有认真研究另一个同样重要的问题：当运算结果出来之后怎样知道它是不是正确，是否是真正想要的。现在由于采用了二进制，这个问题迎刃而解了。方法出奇的简单，因为运算部件是以二进制的方式工作的，它送出的运算结果自然也是用一排导线表示的二进制数。这样，我们可以把小灯泡接在每一根输出的导线上，以此来显示这些输出的比特到底是0还是1（图2.19）

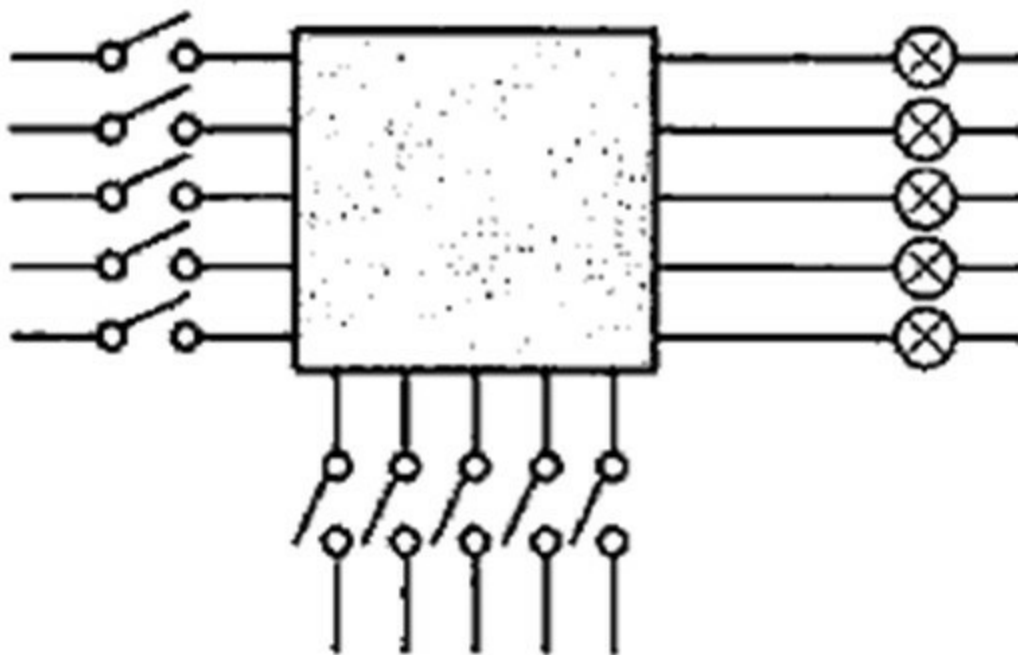


图2.19 通过使灯泡发光，可以直观地看到运算结果

这个办法很好，而且特别有意思，它使得结果能以可视的形式直接被我们用眼睛观察到。当某根导线上没有电时，与它相连的灯泡不亮，代表这一比特是0；当灯泡亮时，表明这一比特是1。如果依次记下这些比特并将其换算成十进制，就能知道结果到底是几

看来二进制与电学还真是有不解之缘，好像二进制是专门为发明电子计算机而量身定做的。

但二进制和电子计算机之间本身毫无关联。二进制论的创建时间大约在1672-1676年，发明者是德国人莱布尼茨，莱布尼茨是伟大的哲学家和数学家，他不但是数理逻辑的开创者，还是与牛顿齐名的数学家，他们两个人相互独立地创建了微积分。尽管现在公认的是莱布尼茨创建的微积分符号要优于牛顿的，而且一直到今天都在使用，但他们谁先发明了微积分的问题上曾经争吵不休，不过在另外一些场合，莱布尼茨很欣赏牛顿，曾经高度称赞牛顿的数学成就“比得上在他之前所有成就的总和”。

尽管莱布尼茨发明了二进制，但这并非是由于他认识到二进制对于计算机的重要性，事实上，尽管他的确曾经热衷于研究如何制造计算机，而且也确实发明了一台机械式计算机，但那台机器根本不使用二进制工作，也和二进制毫不相干。

这次的设计无可挑剔了，至少从技术难度上来说是完全可行的，所以开始着手构造这个运算部件内部的电路了

## 第3章 怎样才能让机器做加法

现在我们已经拥有了二进制的知识，也学会了用开关将电流表示成二进制比特，为了查看运算结果还使用了灯泡。看起来开端不错，现在核心工作就是要清楚这个加法器的内部构造，这不是一件容易的事，我们需要好好规划一下

在开始之前，用机器来算数学题所面临的问题在于，没有生命的东西，如一段电线、一个电池是没有智慧，不会算数学题的。为了制造一个会算数学题的机器，必须先回顾一下我们自己算数学题的方法和过程，然后用零件和电路来模拟这个过程。

理解了这一点后，先从熟悉的十进制加法开始，再看看如何用二进制数做加法

### 3.1 我们是用十进制做加法的

尽管十进制加法很容易，十进制加法的计算口诀是：

0加0等于0；

0加1等于1；

0加2等于2；

.....

9加7等于6，进1；

9加8等于7，进1；

9加9等于8，进1；

下面通过一个例子，比如计算 $15+7$ ，来看看这些口诀是如何运用的  
如图3.1所示，在实际做加法时，要将被加数和加数从右边对齐，  
而开始计算的时候，也同样是从最右边的一位开始

$$\begin{array}{r} 15 \\ + 7 \\ \hline 22 \end{array}$$

图3.1 十进制加法示意图

首先是 $5+7$ ，根据口诀“5加7等于2，进1”，结果得2，并向左产生一个进位，为了防止把这个进位忘记了，上学时老师会告诉我们一个小小的技巧，在左边一列的下面写一个小小的1，表明这只是个进位

现在左边剩下1，因为还有一个进位，所以再次使用口诀“1加1等于2”得到这一列的结果2，至此完成了这道加法题，结果是22

## 3.2 用二进制做加法其实更简单

尽管讨论十进制加法对我们来说显得很轻松，但它不是这一章的重点。现在接着讨论另外一个更重要的话题，它关系着我们的加法器是否能够顺利地制造出来

二进制加法怎么做呢？

$$11101+110=?$$

在做十进制加法时有两点：

- (1) 必须掌握加法口诀；
- (2) 考虑进位。

二进制加法这两点同样重要

十进制加法有一大堆口诀，因为十进制有0-9十个基本数字，加法口诀需要把它们都组合起来

相比之下，二进制加法的口诀比较简洁，因为二进制只有两个基本数字：0和1。所口诀是

0加0等于0；

0加1等于1；

1加0等于1；

1加1等于0，进1

知道了二进制加法的口诀后，让我们实际做一下110+11

如图3.2所示，和十进制加法一样，两个要加起来的数右对齐，然后从最右边的列开始计算

$$\begin{array}{r} \phantom{+} \phantom{1} \phantom{1} \phantom{0} \\ \phantom{+} \phantom{1} \phantom{1} \phantom{0} \\ + \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \hline 1 \phantom{0} \phantom{0} \phantom{0} \phantom{1} \end{array}$$

图3.2 二进制加法示意图

先是0加1，对应口诀“0加1等于1”，这一列的和是1

接着，1加1，对应口诀“1加1等于0，进1”所以这一列的结果是0，同时向左边产生一个进位1

继续向左，这一列是1，还有一个进位，对应“1加1等于0，进1”，



这一列的结果又是0，而且也向左边产生一个进位1，左边除了1个进位没有数字。所以最终的结果是1001

### 3.3 使用全加器来构造加法器

任何一个二进制数都是由一个以上的比特组成的，是一个比特串。为了突出组成它的每个比特，一个二进制数可以表示成（如果它包含6比特的话）：

$$a_5 a_4 a_3 a_2 a_1 a_0$$

在这里 $a_0, a_1, a_2, \dots$ 都是一个比特，是这个二进制数中的每一位。

我们日常生活中通常从1开始编号，但这里是从0开始，而且是从右向左递增，为什么是这种形式呢？电子计算机是外国人发明的，我们研究计算机技术，引进他们的技术资料，自然也就把这个学过来了。

所以如果 $a_5 a_4 a_3 a_2 a_1 a_0 = 110101$ ，那么它们的对应关系如图3.3所示

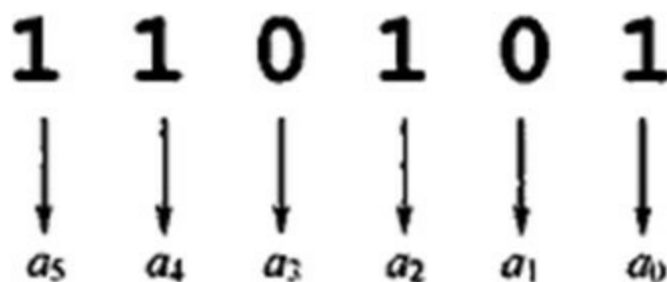


图3.3 二进制数中各个比特的编号方法

为什么要写成这种形式呢？目的是分析在不知道或不用知道两个二进制数是几的情况下，它们是如何相加的，这里面有什么规律。如果可能的话，我们就可以顺利地把加法器制造出来了。

那么来看看随便两个二进制数相加时会怎样，比如：

$$a_5 a_4 a_3 a_2 a_1 a_0 + b_5 b_4 b_3 b_2 b_1 b_0$$

它们可以是任意两个数，先把它们右对齐，如图3.4所示

$$\begin{array}{rcccccc} & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ + & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\ \hline \end{array}$$

图3.4 两个二进制数相加，要和十进制加法一样对齐

因为最先相加的是最右边一列，即 $a_0$ 和 $b_0$ ，所以这里没有其他列的进位，属于单纯的两个比特相加，如图3.5所示，这里有4种可能的情况：

$\begin{array}{r} 0 \\ 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 1 \\ 1 \\ \hline 0 \end{array}$
---	---	---	---

图3.5 不考虑其他列的进位时两个比特相加的4种可能

根据二进制加法口诀，很显然，只有在来自被加数和加数的比特都是1的时候，也只有在这个时候，才会向左边进位，并且结果是0

相比之下，倒数第二列就不是单纯的 $a_1$ 和 $b_1$ 相加，还可能会有后一列的进位，所以这一列实际上就是三个比特相加。当然，如果后一列没有进位，那么情况和图3.5一样

不过，如果最右边一列产生了进位，那么这一列实际上是另外4种可能，如图3.6所示

$\begin{array}{r} 0 \\ 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ \hline 0 \end{array}$	$\begin{array}{r} 1 \\ 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 1 \\ 1 \\ \hline 1 \end{array}$
---	---	---	---

图3.6 存在其他列的进位时两个比特相加的4种可能

综合图3.5和图3.6，就得到了在这一列上做加法时，所以有可能出现的情形，其8种。一列相加的结果可能是0，也可能是1，而且可能产生进位，但进位必定是1（而不会其他数值）

再研究其他列相加的情况已无必要了，原因是除了最右边那一列，不管哪一列相加，情况都和上面讨论的一样，每一列都有可能需要加入前一列来的进位（1），相加的结果可能是0，也可能是1，并且它自己也可能向前一列进位（1）

既然加法都是按列进行的，而且每一列的计算机过程都一样，那么完全可以设计一个电路来完成每一列的相加过程，如图3.7所示

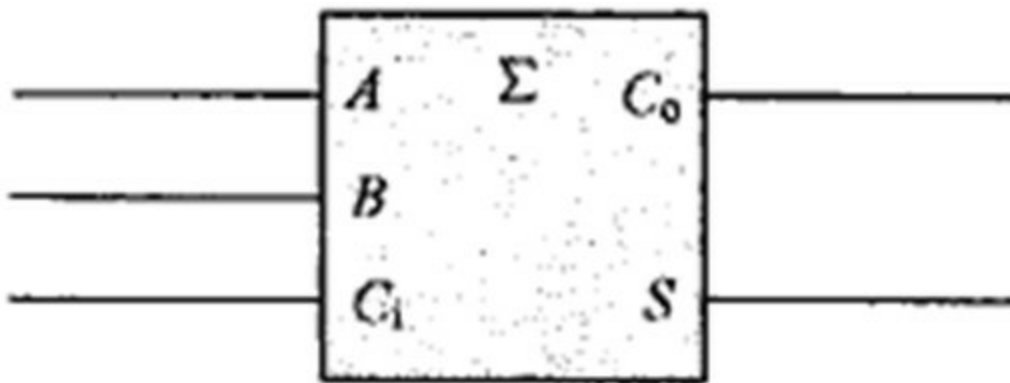


图3.7 全加器示意图

图中，A和B分别是来自被加数和加数的一个比特，它们正好在同一列上； $C_i$ 是来自右边一列的进位； $C_o$ 是本列产生的进位；S是本列的“和”，为了表明这个电路的用途，在图的中间加了一个符号 $\Sigma$ （在数学中，这个符号表示“加”，读音是“西格马”）

这个器件的名称是“全加器”，这不是一个很容易理解的名字，特别是这个“全”字，而且，既然是全加器，是不是还应该还有“半”加器？还真有半加器这东西。但半加器仅仅是把来自被加数和加数的两个比特加起来，产生和、进位，并不考虑从其他列来的进位。换句话说，它只是用电路来实现二进制加法口诀，全加器则不然，它真正实现了二进制加法中每一列的加法过程，所以它才叫“全加器”

有了全加器，解决了二进制加法过程中每一列的计算问题，那么，我们可以组装一大堆全加器，根据被加数和加数的比特数，把它们串联起来组成一个完整的加法电路，图3.8显示了这一过程

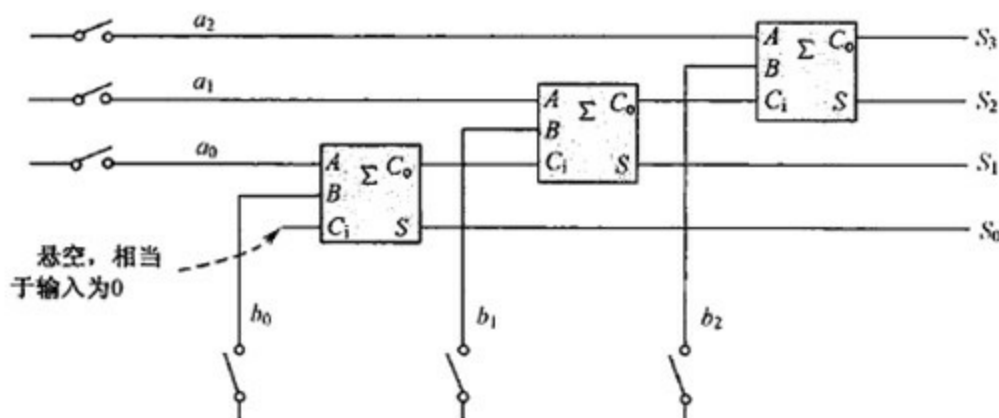


图3.8

图中，参与相加的两个二进制数分别是 $a_2a_1a_0$ 和 $b_2b_1b_0$ ，组成它们的每一个比特可以用开关的闭合与断开来得到。

随着开关的闭合与断开，会得到一些二进制数，比如 $a_2a_1a_0=110$

但也有可能会得到一些看似不那么正常的，比如 $a_2a_1a_0=011$ ，它实际

是11，只不过前面多了个0，和十进制一样，在1左边不管有多少个0，都不会改变它的大小。

因为被加数和加数各自用了3个开关，很明显，参与运算的被加数和加数都只是3比特的二进制数。之所以没有使用比3比特更大的二进制数是因为这样看起来更清楚

像按列做加法一样，3个全加器串联起来，把被加数和加数中位置相同的两个比特相加，输出结果，并将进位传递给下一个全加器，可以看出，第一个（左下角那个）全加器的进位输入端没有使用，意思是“没有进位输入”，或者“从后面来的进位是0”。其余的全加器的进位输入端都和前一个全加速器的进位输出端相连，意思是“前面的，你产生进位了吗？如果有，我得加上它”。 $S_2S_1S_0$ 是两个二进制数相加后的最终结果。 $S_3$ 是最后一个全加器产生的进位，由于这是最后一个全加器，所以它的进位也是最终结果的一部分。尽管 $100+1$ 的结果是3比特

（101），但 $100+100$ 却产生4比特的结果（1000），最左边的1纯粹是由进位产生的

仔细思考，还可以发现这个加法器很容易进行扩充以计算更大的数（这样会更实用），唯一所要做的就是在此基础上再串联更多的全加器。

在所有记数方法中，二进制不见得是最实用的，用于表示数太长，但计算机需要它，因为它简单，它的一切只有0和1。简单意味着具有较少的运算规则；较小的运算规则意味着设计不太复杂；不复杂的设计又意味着可以用很少的材料来制造，还能保证机器工作的可靠性。

要造一台机器来计算加法，全加器无疑是最基础的零件了，但这个全加器到底应该如何构造呢？看起来我们离目标不远了，应该很快就能揭开最终的谜底了

事实上，还差得远呢。要想知道全加器的内部都是些什么，它是如何把3个比特加起来的，还必须回到过去，了解电与磁的历史，以及先哲们创立的逻辑学，看能不能从中得到一些启发

## 第4章 电子计算机发明的前夜

计算机的发展虽然在几千年前就开始了，但只有在最近几十年来在计算机方面的研究突飞猛进，这并非人类突然变聪明了，而是得益于人类在电磁学、数学、逻辑学等领域里取得的成果。

尽管从表面上看电子计算机是神奇的、智慧的，但掩盖不了它实质上只是一种普通电器的事实，特别是随着技术的成熟、产量的增加，以及价格的降低，电子计算机越来越普遍了，作为一种前所未有的、特殊的电器，它肯定比灯泡复杂，需要更多的零件，这也意味着，要想清楚它内部到底是怎样运作的，仅仅靠掌握一些简单的电学知识是不够的，还必须了解另外电磁学的历史，没有它提供的理论知识和电子零件，电子计算机的发展也就失去了最原始的基础。

一定要回到过去，重温科技史上那些有趣的事件和激动人心的瞬间，弄清楚都是什么能够让我们用上电子计算机，并通过电子计算机阅读新闻、编辑文章、打印表格、以最快的速度 and 远方的朋友交流.....

那么，先从电磁学的历史开始吧

## 4.1 电能产生磁

制造能够自动计算的机器并不是人类唯一的梦想，自古以来，人类的梦想很多，比如飞翔、长生不老、在遥远的地方看到自己的亲人或互相交谈……而商人更希望有办法能及时与千里之外的同伴进行联络，让他们知道他所在的地方需要茶叶、蚕丝……。此外当有敌人入侵时，前线需要有办法能快速通知后方。在古代他们能想到最好的办法就是点燃烽火、狼烟，通过接力的方式来传递这些信息

如果速度不是问题的话，那从前没有任何电信设施的年代，使用邮政可能是一个好办法。但实际上，速度恰恰是人类最关心的问题。但这个问题也只有在电走进人类的生活后才有可能解决

我们知道，电流的速度每秒30万千米，所以如何使用电来传递信息就成了人们需要认真研究和思考的问题。毕竟，在人们对它进行改造以适合传递信息之前，它只能用来在另一个地方生成光和热。看起来我们还有一段路要走。不过，在找到解决办法之前，我们应该先来研究一下钉子 – 生活中最普通的钉子

一个普通的钉子没有磁性，不能吸引别的东西，但是要让它具有磁性也不难，只需要像图4.1那样用一根导线绕在上面

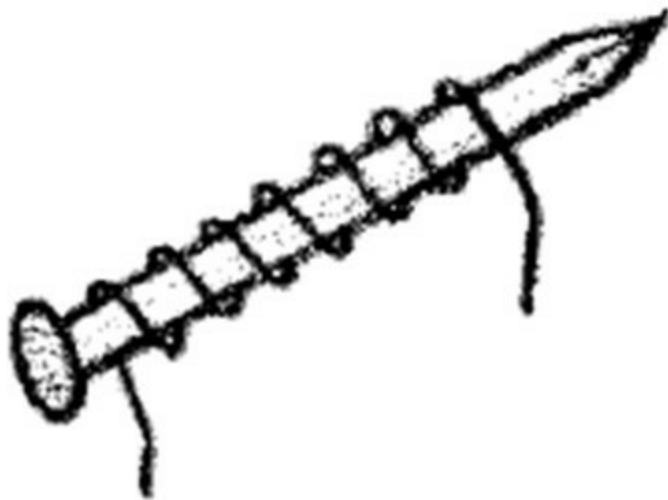


图4.1 用电线绕在铁钉上就能制成电磁铁

**注意：** 不要使用裸线，会因为与铁钉接触短路而失去效果。

匝 – 意思是“周”，环绕一周称为一匝。

如果有条件，可以使用较粗的漆包线，漆包线顾名思义就是导线表面包裹着一层漆皮。应该在铁钉外面绕多少匝，这个没有规定，当然匝数越多效果越好。它的名称是电磁铁

现在我们都知道，当一根电线有电流通过时，就会在它的周围产生微弱的磁场，电能产生磁的现象叫做电流的磁效应。毫无疑问，这是一个伟大的发现，这个荣誉属于丹麦物理学家汉斯·克里斯蒂安·奥斯特（Hans Christian Oersted）

1820年，一个偶然的机会，奥斯特发现当电路接通时，离电线很近的磁针（相当于指南针，在地球这个大磁场的作用下会指南指北）会发生偏转。由于这一发现，奥斯特在连续进行了一段时间的实验和研究后，于当年发表了题为《磁针电抗作用实验》的论文，向科学界公布了他关于电流磁效应的发现。

奥斯特的偶尔发现说明了一个事实，那就是电流可以产生磁场，不用怀疑，你家里的电线周围就有磁场，耳机线周围也有磁场，但这些磁场很微弱。如果把电线绕起来，这就相当于很多小磁场的叠加，磁力就会大大增强，也就是说，匝数越多，磁场越强

电流能产生磁场这种现象吸引了很多人制作了各种各样的新鲜玩意儿，它们大部分实际上就是电磁铁，那时人们制作电磁铁的兴趣是如此高涨，据说1831年，美国物理学家约瑟夫·亨利制作了一个体积并不是很大的电磁铁，能吸引重达1000kg的铁块。他制造的电磁铁之所以性能这么好，原因在于为了使线圈能绕得密集一些，他采用了用细纱包裹的绝缘导线（当时还没有给导线上漆制造漆包线的技术），这样线圈的匝与匝之间可以紧挨在一起而不用担心短路。

亨利从小就是个有志向的人，不过麻烦在于他总能因为各种原因找到新的志向，一开始，他在剧本创作方面找到了属于他的人生意义，可是不多久，也许是因为这样无法维持生计，他又到钟表修理铺当学徒，维修指针和发条。最后因为读了一本自然科学方面的书，又激发了他上学的兴趣，后来当过教授，教过数学、哲学，最后成为大学院长。



## 4.2 继电器和莫尔斯电码

电学的发展还在继续。

电可以产生磁，可以用来制作电磁铁，这在当时是非常新奇的事情。不久，历史上第一个尝试用电来进行远距离通信的人出生了

撇开电磁铁不谈，单单是发现电流速度很快这一点，有人就想到了它可以用来传递信息。那个时候，想用电流在两个地方传递信息的人不止一个，但唯一获得成功的是塞缪尔·摩尔斯（命名塞缪尔·芬利·布里斯·摩尔斯，Samuel Finley Breese Morse）。1832年秋天，他乘船从法国到美国，遇到了一位美国医生 - 杰克逊，这个医生的医术如何没有人知道，但这个医生居然懂电磁铁的原理和制造技术，他甚至知道线圈匝数越多，电磁铁的吸引力越强。旅途漫漫，为了打发时间，杰克逊拿出电磁铁给大家变起了魔术，可以想象，当人们看到像钉子这类铁的东西，在医生的命令下说让它吸住就吸住，让它掉下来就掉下来，摩尔斯也在人群中看热闹

摩尔斯（1791-1872）生于美国一个牧师家庭，1810年毕业于耶鲁大学，早期曾从事印刷和绘画。作为一名画家，他是成功的，摩尔斯曾两度赴欧洲留学，在混肖像画和历史绘画方面成了当时公认的一流画家，1826年至1842年任美国画家协会主席

摩尔斯向杰克逊医生请教电学方面的知识，回到美国后，他决心改行，那一年，他已经41岁了。他全身心地投入到研制电报机的工作中去，他拜著名的电磁学家亨利为师，从头开始学习电磁学知识，他买来各种各样的实验仪器和电工工具，把画室改成实验室，他设计了一个又一个方案，绘制了一幅又一幅草图，进行了一次又一次的试验，但得到的是一次又一次的失败。他分析了失败的原因，认真检查了设计思路，发现必须寻找新的文法来发送信号。1836年，他终于找到了新方法，他在笔记本上记下了新的设计方案：“电话只要停止片刻，就会出现火花。有火花出现可以看成是一种符号，没有火花出现是另一种符号，没有火花的时间长度又是一种符号，这三种符号组合起来可代表字母和数字，就可以通过导线来传递文字了。”

摩尔斯发明的是一种称为电报机的东西，由不在一个地方的两个装置组成，用很长的电线连接起来，如图4.2所示

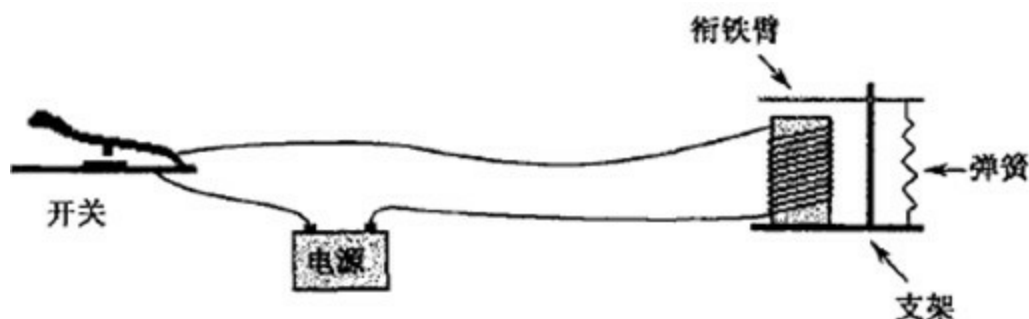


图4.2 莫尔斯电报机示意图

一个开关，通常称为按键，可以控制电流的通断，只是与我们常见的开关在形状上有很大的差别，但作用是一样的，因为要长时间在按键上反复操作，所以在设计上强调方便和舒适性，而在电线的另一端，连着一个电磁铁，这样，通过按下或松开按键，就能控制磁性的有无。在电磁铁的上方，有一个长长的铁片（衔铁臂）安装在支架上，它可以上下自由活动，平时，也就是电磁铁没有通常产生磁力的时候，衔铁臂被一根弹簧拉着，以免与电磁铁挨在一起，这样，一旦开关闭合，衔铁臂就会被电磁铁吸引；当开关松开，电磁铁失去磁性，衔铁臂又在弹簧的拉力下回到原始位置

这个装置没有什么新奇的地方，但如果在衔铁臂上安装一支笔，并在笔的下面放一卷纸带，纸带匀速移动，当按下按键并迅速松开，会在瞬间使电磁铁产生一个吸合与释放的动作，结果是笔尖在纸上打出一个“.”，如果按键按下的时间稍长一点，那么笔尖会在纸上留下一条线“-”，这称为“划”，连续地按动按键，就会在纸上留下一串由点和划组成的图案，像这样：

·-·-·-·-·-·-·-·-·-

这正是莫尔斯所发明的装置 – 电报机的核心原理。不过，要想让这个装置有用，最关键的是需要一张发送方和接收方都能理解的电码表（电报通信中用来代表文字、数字、标点等的符号），在这张表里，用点和划的组合来表示从A-Z的26个英文字母以及从0-9的十个数字

# 摩尔斯码



摩尔斯电码（Morse Code）由两种基本信息和不同的间隔时间组成：短促的点信号·（读Di）；保持一定时间的长信号—（读Da）。间隔时间：Di,1t;Da,3t;DiDa间，1t;字符间，3t；字间，7t

虽然摩尔斯发明了电报，但他缺乏相关的专业技术，他与阿尔弗雷德·维尔（Alfred Vail）签订了一个协议，让维尔帮助自己制造更加实用的设备。阿尔弗雷德·维尔构思了一个方案，通常点、划和中间的停顿，可以让每个字符和标点符号彼此独立地发送出去。他们达成一致，同意把这种标识不同符号的方案放到摩尔斯专利中，这就是我们现在熟知的美式摩尔斯电码，它被用来传送了世界上第一条电报

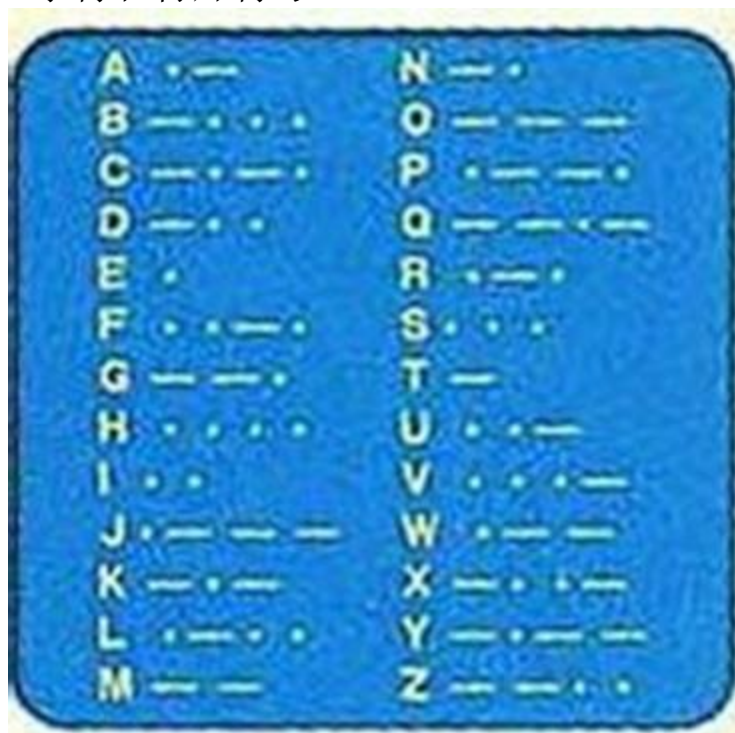
这种代码可以用一种音调平稳时断时续的无线电信号来传送，通常被称为“连续波”（continuous wave, CW）。它可以是电报电线里的电子脉冲，也可以是一种机械或视觉的信号（比如闪光、旗语）

摩尔斯电码特指两种表示英语字母和符号的编码方式：美式摩尔斯电码被使用了在有线电报通信系统；国际摩尔斯电码则只使用点和划（去掉了停顿）

## 电码分类

美式摩尔斯电码

作为一种实际上已经绝迹的电码，美式摩尔斯电码使用点、划和间隔来表示数字、字符和特殊符号



A . —	N — .
B — . . .	O — — —
C — . — .	P . — — .
D — . .	Q — — — .
E .	R . — .
F . . — .	S . . .
G — — .	T —
H . . . .	U . . —
I . .	V . . . —
J . — — —	W . — — —
K — . —	X — . . .
L . — . .	Y — . — —
M — —	Z — — . .

这种摩尔斯电码的设计主要是针对地面电报员通过电报电线传输的，而非通过无线电波。

这种古老的、交错的电码是为了配合电报员接听而设计了，不像现在可以从扬声器或耳机中听到电码的音调，那时只能从这些早期的电报机的机械发生装置听到DiDa声，甚至是从发送电键接听，发送电键在不发送信号时被设置为被动模式，负责接收发声

国际摩尔斯电码

国际摩尔斯电码是由普鲁士的弗雷德里希·克里门斯·盖尔克（Friedrich Clemens Gerke）在1848年发明的，用在德国的汉堡（Hamburg）和库克斯港（Cuxhaven）之间电报通信。1865年之后在少量修改后由国际电报（International Telegraphy）大会在巴黎标准化，后来由国际电信联盟（ITU）命名为国际摩尔斯电码



因为摩尔斯电码只依靠一个平稳的不变调的无线电信号，所以它的无线电通讯设备比其他方式的更简单，并且它能在高噪声、低信号的环境中使用。同时它只需要很窄的带宽。

历史上第一份长途电报是在1844年5月24日发出的，这表明莫尔斯的发明已经具备了实用性，不过如果线路太长，电阻会变大，这样在电报线路的另一端，微弱的电流将不能使电磁铁正常吸合，电报机也就没有作用了

所以人们在每隔一段距离设置一个电报中转站，派人在那里接收电报，然后再原样通过下一条线路再发送一遍，就这样一段一段地接力传递

这当然是个好主意，但问题是人没有机器精确、可靠，解决之道是使用继电器（electric relay）。继电器顾名思义是给线路续电的，也就是说，当线路上电流过小的时候，适时补充。原理如图4.3所示

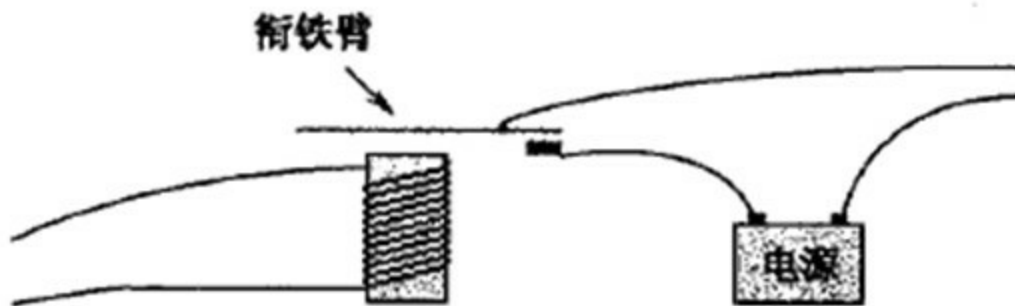


图4.3 继电器的本质是“续”电

这是一个简化的示意图，省去了支架之类的东西，为的是能看得更清楚。它的主体是一个电磁铁，不过衔铁臂下面多了一个金属触点，现在，分别从衔铁臂和金属触点上引出两根线，并串接一个电源，把这两根线作为另外一条电报线路架设到其他地方，注意，电源并不是继电器的组成部分

这是一个奇怪的装置，它应当被放在远离电报发送端，但还可以保证电报信号能让电磁铁正常吸合的地方。当发送方发送一个“.”的时候，衔铁臂也短暂吸合一下，把另一条线路接通；如果发送方发送一个“-”，衔铁臂吸合的时间也和发送方保持一致，让另一条线路上同样发送一个“-”

今天继电器得到广泛的应用。从电视机、洗衣机、电冰箱到工厂里的大型工业设备。

继电器的符号如下图：

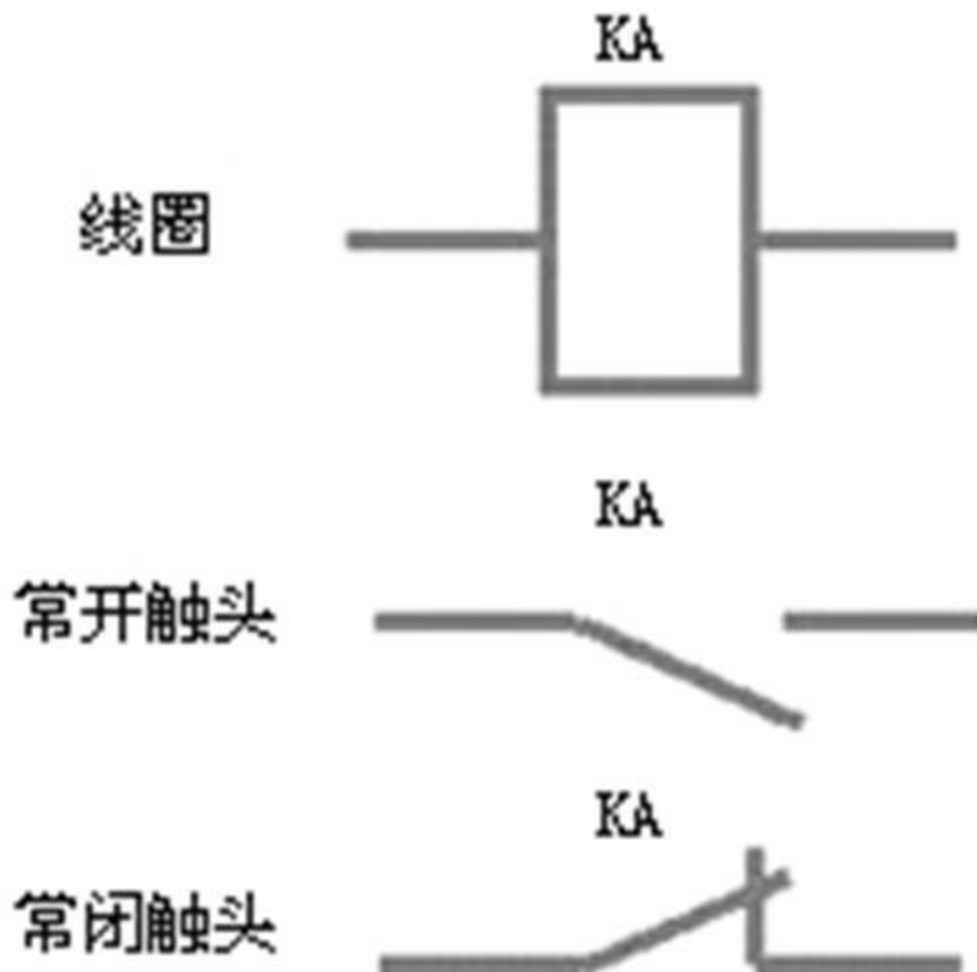


图4.4 继电器符号

图中的方框代表电磁铁，而在下面紧挨着它的是一个衔铁开关，同时，图中也表明了继电器实际上可以按工作状态分为两种：常开触头平时处于断开状态，只有电磁铁加电时才吸合接通；常闭触头正好相反

# 电报的历史

电报是通信业务的一种，在19世纪初发明，是最早使用电进行通信的方法。电报加快了消息的流通，是工业社会的一项重要发明。早期的电报只能在陆地上通讯，后来使用了海底电缆，开展了越洋服务。到20世纪初，开始使用无线电电报。电报的主要用来传递文字信息，使用电报技术传送图片称为传真

## 电报之前的通信方法

在未发明电报之前，进行长途通信的方法主要包括：驿站传递、信鸽、烽火（狼烟）、摆臂式信号机（semaphore）等等。

## 电报的发明

欧洲的科学家在18世纪逐渐发现电的各种特质，同时开始有人实验使用电来传递消息的可能。早在1753年，一名英国人摩尔逊便提出使用静电来传递消息，他的设想是使用二十六条电线分别代表二十六个英文字母，发信息的一方按文本顺序在电线上加上静电，接收的一方在各电线上接上小纸条，当纸条因静电而升起时，就传递了消息，但这种机器需要的导线太多，设置复杂，并且静电感应的距离有限，因此这项发明没有得到推广。

1804年，西班牙的萨瓦将许多代表不同字母和符号的金属线浸在盐水中，他的电报接收装置是装有盐水的玻璃管，当电流通过时，盐水被电解，产生小气泡，根据这些气泡辨识出字母，从而接收到远处传过来的信息。但萨瓦的电报接收机可靠性很差，不具实用性。后来俄国科学家许林格设计了一种用8根电线的编码式电报机，并且取得实验上的成功，但由于需要的导线太多，依然难以达到实用。

这些电报装置虽然没有得到最终应用和推广，但它们提供以试验基础，随着电磁学理论的不完善，电学的进一步发展，一根导线的电报机在摩尔斯那里诞生了

1843年，塞缪尔·摩尔斯用国会赞助的3万美元建起了从华盛顿到巴尔的摩之间长达64公里的电报线路，1844年5月，他在华盛顿国会大厦最高法院会议厅里，用他从1937年发明出来并不断完善的电报机，向巴尔的摩发送了世界上第一封电报，内容是《圣经》中的一句话“上帝啊，你创造了何等的奇迹！”

自此之后，这种“闪电式的传播线路”迅速发展，电报本身并不是大众传媒，但它为大众传媒提供了快速有效的传播手段。

电报收发报机主要有两类



## 摩尔斯电报机

摩尔斯电报机由报务员用电键发送。电键实际上是一个易于持久操作的开关，由按键时间的长短来决定点或划。收报机在匀速前进的纸带上划出点或划，或用声响等方法来显示出所收到的电码，报务员由此录出字符电文。这种方式称为人工电报。人工电报虽然是一种早期方式，速率较低，但设备简单，对传输电路的要求不高，工作比较稳定

1858年出现了用摩尔斯电码的自动收发报机。发报局用专用的凿孔机先在纸带上凿出与电文字符相对就把圆孔，再把这种凿孔纸带在摩尔斯电码自动发报机上发送，收报局则用波纹收报机在纸带上录出点划电码，其速率可比人工电报快几部到20倍左右

## 五单位电码电报机

电传打字机是采用五单位电码的电报机。它出现于20世纪20年代，从30年代初起得到广泛应用，发展较快。电传打字机在外形上与普通打字机相似，由于发或收任一字符所需的时间相同，可简化机器的设计与构造。电传打字机主要包括发报键盘和收报打印字两个部分，因此兼有发报和收报功能。发报部分由装有打字字键的键盘发送出五单位电码的脉冲信号；收报部分则依照收到的电码信号在纸页或纸带上打印出字符，省去了电码和字符之间的人工译码的工作，因此效率高且使用方便

电传打字机通报时两端的发报机和收报机必须同步工作，每传送一个字符同步一次。因此要在每个字符的五单位电码第一脉冲前加一个空号性的起动脉冲，其长度与一个信号脉冲相同，而在第五脉冲之后要加上一个传号性的停止脉冲，其长度为1.5个信号脉冲。所以每个字符的实际长度为7.5个脉冲。起动脉冲和停止脉冲的作用是使电传打字机每发、收一个字符时机器起动和停止各一次，以保证发报机和收报两端的电传打字机在通报时能得到同步，因此电传打字机也称为起止式电报机。

在收报印字器件的每个字位上除字母外还有一个数字或符号，按下发报键盘上的“字母位”机能键后，能使收报印字部分只印出字母；而按下“数字位”机能键后则收报部分只印出数字符号。印出的字母或数字符号各有26种，因此电传打字机共可印出52种字符

传真电报简称传真。使用传真可直接传送发报人文件、图形、表格、照片等。它之所以受到欢迎，主要在于通信速度快、操作简便，对方只需一台传真机就能接收与原样相同的复印件。其传输方式分直流电报和载波电传传输。若实施电报通信，一定不能缺少两部分设备，一是电报通信的终端设备，如人工电报机、电传打字机、五单位自动发报机等；二是电报通信的传输设备，如通信线路、载波电报机、无线收发信机等。

电传打字机包括两大类，一是机械式电传打字机，依靠电动机带动一系列机械动作来完成接收或发送信号；另一类是电子式电传打字机，它的发报、收报及各部分动作的协调由电子逻辑电路控制完成。

真正首条投入运营的用电传递消息的线路于1839年在英国出现，它是大西方铁路（Great Western Railway）装设在两个车站之间进行通信。这条线路长13英里，属指针式设计，由查尔斯·惠斯通（Charles Wheatstone）及威廉·库克（William Cooke）发明，两人并为发明在1837年取得英国的专利。

在美国，塞缪尔·摩尔斯（Samuel Morse）在几乎相同的时间发明了电报，并在1837年在美国取得专利。

摩尔斯还发展出一套将字母及数字编码以便使用电报发送的文法，称为摩尔斯电码。

## 电报的发展

初期的电报只能通过架在陆地上的电线（land wire）进行通信。最早的电线属于单线式，需要通过地面形成回路，传送距离有限。到了1850年，英国人约翰和雅格布·布雷特（Jacob Brett）兄弟俩估法国的格里斯-奈兹海角（Cape Gris-Nez）和英国索兰海角（Cape Souter-land）之间的公海里用“巨人”号拖船在英法两国之间的多弗海峡铺设了第一条海底电缆，但只发了几份电报就中断了，原因是一个渔夫用拖网勾起了一段电缆，并截下一节向别人夸耀这种稀少的“海草”标本，说里面装满了金子。

1958年，第一份海底电缆电报横越大西洋，这条大西洋海底电缆于1857年8月7日从爱尔兰海岸瓦伦西亚（Valentia）开始铺设，8月17日海底电缆在12000英尺深的水下崩断。1858年7月28日深夜，两只铺设缆船再次在大西洋中部相会，拼接好电缆。8月5日，总长为3240公里的电缆铺设完毕。凌晨2：45，第一份海底电缆电报横越大西洋。8月12日，美国和英国之间播发海底电缆电报，9月3日1点，由于报务员的错误导致电缆绝缘击穿而损坏。首条大西洋海底电缆在1866年成功投入使用。至于横越太平洋的海底电报电缆，直到1902年才完工。

到了19世纪90年代，各地仍然要经过电线来传送电报，尼古拉·特斯拉（Nikolas Tesla）等科学家在这时开始研究以无线电发送电报，1895年，意大利人马可尼（Guolielmo Marconi）首次成功收发无线电电报。1898年，他成功地进行了英国到法国的无线电电报的传送。1902年首次以无线电进行横越大西洋的通信，无线电电报的发明使移动通信成为可能，配备无线电的远洋船只就算在海洋上仍然能与陆地保持通信。

## 4.3 磁也能产生电

人类的优点是擅长站在别人肩膀上发现新的问题，在奥斯特发现电流能产生磁场后，人们想到既然电流能够产生磁场，那么反过来，磁场能不能变成电流呢？

当时有很多人在研究这个课题，如瑞士的物理学家科拉顿。1825年，科拉顿做了一个实验：先制作一个大的空心线圈，把它的两端接到一个电流计上，将一块磁铁插入线圈中，观察电路中是否有电流产生，如图4.5所示。



图4.5 电磁感应实验

电流计是一个检测电路中有没有电流通过的装置，当有电流通过时，它里面的指针会发生偏转

这本应该是一个非常成功的磁产生电实验（直到现在向学生传授磁产生电的知识时还是使用这种方法，可见这种方法非常简单有效），但科拉顿犯了一个愚蠢的错误，他把电流计放得很远，这样，每次把磁铁

插入线圈里之后，再跑过去看电流计的指针是否发生了偏转，这样他就错过了观察电流计指针发生偏转的时机 – 事实上，在磁铁插入线圈的过程中，电流计做出了反应

接下来办到法拉第了

迈克尔·法拉第（Michael Faraday）出生于英国，他的父亲是一名普通的铁匠，少年时代的法拉第当过报童，后来干书籍装订。他爱看书，喜欢做实验，曾经听过戴维爵士的科学讲座。戴维爵士比法拉第大13岁，是著名的化学家，也是当时英国皇家学会的主席，经常在学院里举办科学讲座，据说他善于把复杂的科学问题通俗化，吸引了很多人。

年轻的法拉第渴望从事科学研究，他把讲座听到的内容认真整理成笔记，中间还加上自己的见解，他把这本笔记连同一封信寄给了戴维爵士，在信中表达了自己希望能够到皇家学院从事科学研究的强烈愿望，戴维爵士很感动，没多久，他如愿以偿地来到皇家学院

为了研究磁如何产生电，法拉第花了十年时间，之所以这么久，是因为在这十年里他并不完是在研究这个，有时他是戴维爵士的助手，帮着做一些化学实验，如果戴维爵士外出考察，他还得充当仆人或跟班的角色。有时他自己也研究一些化学课题，1825年，他还发现了苯

另一方面，在那个年代，人们的想法很朴素，他们觉得磁产生电就是将一根导线静静地放在一块磁石上，然后在旁边看是否能产生电。为了做实验，法拉第用一个大铁圈，在两边绕了两个线圈，如图4.6所示

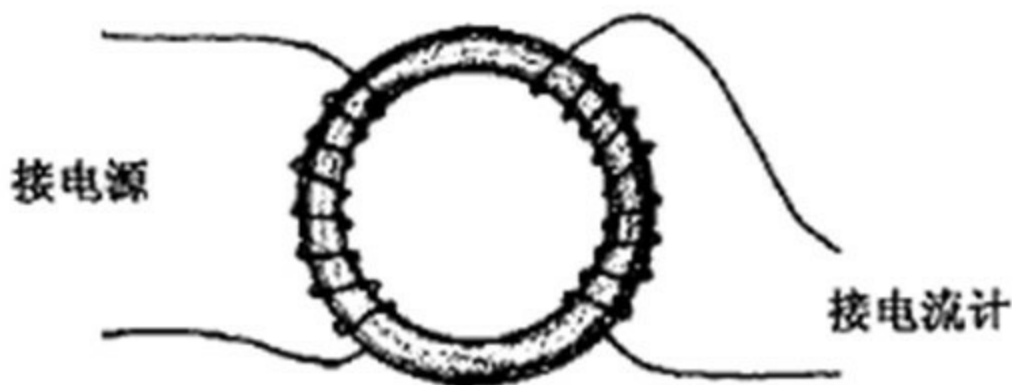


图4.6 法拉第用来将磁变成电的装置

这两个线圈各有各的用处，左边的线圈接开关和电源，实际上是把整个大铁圈变成了一个电磁铁，当闭合开关时，线圈中有电流通过，大铁圈就变成了一个电磁铁，产生磁场；右边的线圈接电流计，按法拉第的想法，如果这个线圈在电磁场的作用下产生了电流的话，电流计就应该发生偏转，观察者就可以发现这个现象。

1831年的一天，当法拉第像往常一样做这个实验时，不知道为什

么，这一次他在给左边的线圈通电时，在电源接通的一瞬间，电流计指针晃动了一下。他重复做这个实验，确保不是自己看花了眼，他终于发现，只有当导体在磁场中运动时，才能产生电流（即磁场的变化在导体中感生出电流）。磁生电也是一种能量转换

## 4.4 电话的发明

莫尔斯的电报只能传递文字信息，而且还得用电码表翻译出来才能知道是什么内容。人们想能不能用电线和电流来传递声音呢？这就需要知道我们是如何听到声音的

声音本质上是一种振动，它可以通过空气或木头这样的介质传播，所以在没有空气的地方，如真空中，声音是无法传播的。当用力敲鼓、锣的时候，鼓面、锣不停振动（哆嗦），推拉四周的空气跟它一起振动（哆嗦），这样声音就传播出去了

我们耳朵内部有一张分隔中耳及外耳的薄膜，解剖学上叫耳膜或鼓膜。当声音到达的时候会导致这层膜也随着振动（观察一下音箱的喇叭单元），而且振动的形式与声音的来源（如鼓、锣，这叫做音源、声源）相同，这样人就听到声音了

当你向池塘中扔一块石头时就形成了波。声音也是一种波，不同的声音有不同的波形，找一把尺子，将它的一端按在桌子上，扳动另一端，此时你听到了什么？

尺子能发出声音是因为它产生了振动，为了清楚它是如何振动的，可以取一块玻璃，用烟熏黑，然后将正在振动的尺子轻轻地与它的表面接触，同时移动玻璃来模仿时间的流逝，这样就记录下了尺子振动的形状，如图4.7所示



图4.7 物体振动时

人耳之所以能听到声音，是因为鼓膜将声波的振动转换成了生物电。生物电刺激大脑中负责听觉的区域。与此相同，为了用电流来传递声音，也需要将声音转化为电流，这就要用到一个叫话筒的东西

话筒的构造很简单，主要是一个线圈和一个磁场，线圈位于磁场中，并与一个纸片或塑料片相连，说话或唱歌时，由于声波的振动，纸

片也被迫振动，从而带动线圈在磁场中运动，并产生强弱不同的电流

话筒产生的电流，其波形和产生它的声波一致，所以通常称为音频电流，为了能听到远处的人在说什么，还要制造一种东西把电流还原成声音，在电学上，这种东西就是扬声器，俗称“喇叭”

其实话筒本身就可以当扬声器用。当音频电流通过线圈，线圈会产生或强或弱的磁场，线圈本身就位于一个磁场中，两个磁场相互作用，不是互相吸引就是互相排斥，线圈是可动的，而磁体是固定的，结果是线圈带动纸片随着音频电流的变化而运动，从而使外部空气也随之振动，我们就听到声音了

第一个发明电话的人，按比较公认的说法是美国人贝尔（1847-1922），他于1876年申请了电话专利权，不过他的方法有个明显的缺陷，就是话筒产生的电流很微弱，那时还没有发明将微弱电流放大的装置，要打电话非得大声说话，所以后来爱迪生发明的碳精送话器更流行。

与贝尔发明的话筒不同，爱迪生发明的这个装置像个小碗，中间填满了用优质无烟煤提炼的碳精砂。在碗口上有一层金属膜，可以导电，但主要用来接收声波的振动。这个装置不能自己产生电流，所以需要串接一个电源，一头接在碗上，一头接在金属膜上。当对着说话时，由于金属膜的振动，导致碗内的碳精颗粒随着声波的变化时而紧密、时而疏松，从而使这个装置的电阻不停地变化，结果，整个电路的电流随之不停地发生变化，就是说，声音已经被转化为电流了

## 4.5 爱迪生和交流电

爱迪生总共有一千多项发明，包括电灯。

爱迪生出生于美国，祖先是荷兰人

爱迪生即是科学家又是商人，计划用他的灯泡点亮整个世界。随着灯泡的生产，爱迪生的商业事业迅速发展起来。他的公司在城市里铺设供电线路，引入千家万户，从发电厂里出来的巨大电能使一只只灯泡发光。但也正是在这个时候开始，一场以科学的名义而发动的商业战争即将拉开帷幕

爱迪生的供电系统采用的是直流电（当用电池给灯泡供电时，电流总是按一个固定的方向流动，这就是直流电）

在当时，爱迪生的发电厂很难把电输送到很远的地方，这并不是直流电本身的问题，而是因为想要远距离传输电必须克服一个问题：导线的电阻，而且供电线路越长，意味着电阻就越大，电会在到达目的地之前大量损耗。

实践证明，提高所要传输的电压，可以降低电在供电线路上的损耗。遗憾的是，这对爱迪生的供电系统不适用，因为要是这样的话，意味着供电线路末端的灯泡会烧毁。这时特斯拉建议爱迪生对现有的供电系统进行改进，以解决不能远距离输送电力的问题。而且他认为这也是一种更经济的做法，但爱迪生没有采纳他的建议

特斯拉（1856-1943）生于南斯拉夫，后加入美国国籍，他的父亲是一位牧师，而他从小就对电学有深厚的兴趣。特斯拉一生中有大量的发明和创新，有很多成了现代发明创造的技术基础。但他终生贫困，很少有人知道他的名字，几乎从来没有得到过与其创造相匹配的尊敬与荣誉（唯一例外的是现在国际上用他的名字作为磁感应强度的单位，这是一个衡量磁场强弱的物理量）

1884年，爱迪生电灯公司的欧洲公司向爱迪生本人推荐了特斯拉，当时特斯拉28岁。但这两个人都不太喜欢对方，爱迪生喜欢不停地做实验，而特斯拉侧重于理论计算，这使得他的工作有时很有成效，他甚至说“如果爱迪生需要从草垛里找一根针，他会马上像勤奋的蜜蜂一根根地检查稻草，直到他发现自己要找的东西，看到这样的做法，我感到非常遗憾。因为我知道，只要一点点理论和计算，他就能省去百分之十的力气。”而且据说这段话被刊登在报纸上

早在和爱迪生见面之前，特斯拉就开始对一种叫“交流电”的事物发生了兴趣。与直流电不同，交流电的方向和大小都是不断变化的，要想了解直流电和交流电有哪些不同，使用图形可能是最直观的方法，也是



工程上常用的方法

首先来看看直流电，它的典型例子就是干电池，找一节干电池，每隔一段时间（比如1秒钟）测量一次电压，这样就能得到一组数据：

第1秒	1.5V
第2秒	1.5V
第3秒	1.5V
第4秒	1.5V
第5秒	1.5V

.....

如果将这些数据绘成图形，通常要使用直角坐标系，创立它的人是数学家笛卡尔，所以也称直角坐标系为笛卡尔坐标系。直角坐标系由水平和垂直且有方向的两条直线组成一个平面，这两条直线称为 X轴（横轴）、Y轴（纵轴）。现在我们用X轴代表时间，Y轴代表电压，用一个一个的“点”把电压和测量的时间关联起来，如图4.8(a)所示。现在测量电压的时间间隔是秒，假如继续缩小每次测量的时间间隔，而且这个间隔足够小的话，这一个个点会挤在一起，形成一条直线，如图4.8(b)所示，这就是直流电的电压图像

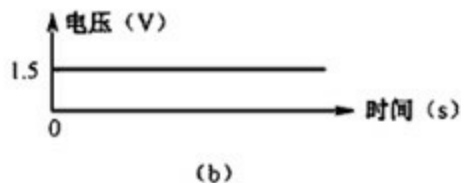
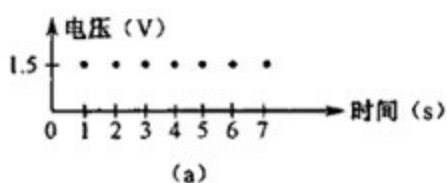


图 4.8 直流电的电压图像

再来说说交流电，在第1章中我们说过，就算是发电厂也不是凭空造出电来，是通过磁产生电（电磁感应），交流电通常就是在大型发电厂里用电磁感应的方式产生的。

为了产生交流电，需要把导体放到一个磁场中，如图4.9(a)所示，在磁铁的两极之间放有一根导线，图4.9(b)是该装置的整体截面图，其中带有箭头的线条是磁力线（磁力线是虚拟的）

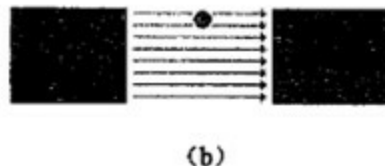
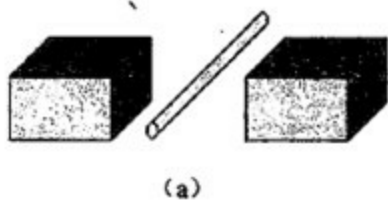


图4.9 导体在磁场中的两种视角

需要指出的是，发电厂决不会用一根导线放在磁场里发电，这产生不了多少电。实际上，是一个绕的很多匝的巨大线圈，外加一个磁力很强的大磁场组成。

为了持续产生电，最好的办法是让导线在磁场中不停地旋转 – 用物

理上的术语来说 – 做圆周运动，这样做有一个好处，那就是可以方便地用水轮或风车驱动。总之，圆周运动肯定是最自然的，要绘制交流电的图像，只需要在导体旋转一周的过程中找几个时机测量一下就行了，如图4.10(a)所示



图4.10 导体在磁场中的运动轨迹及其随时间变化的电压值

乍看起来，电磁感应很简单，随便拿根棍子在磁场中搅和搅和就能产生电。实际上，法拉第发现电磁感应并不简单，在①、⑤处，导体的瞬间运动方向是水平的，与磁力线平行，此时不产生电压，即电压为零；当导体转动到②、④处时，瞬间运动方向与磁力线呈一个角度，此时能产生电压，但并不是很高，角度越小，产生的电压越小；只有在③处，由于导体的瞬间运动方向与磁力线垂直，所以产生的电压最高。图4.10(b)中是我们记录的电压变化情况

从⑤开始，导体经过⑥、⑦、⑧处，最后回到①，这个过程与上半周相同，可以想象，它们的电压变化情况也一样，但令人吃惊的是，电压的极性却和上半周相反。换句话说，突然颠倒了。导体从①到⑤是向下运动的，而从⑤回到①却是向上运动的。感应电压的极性取决于导体的运动方向，所以，导体在磁场中旋转一周所产生的电压如图4.11(a)所示

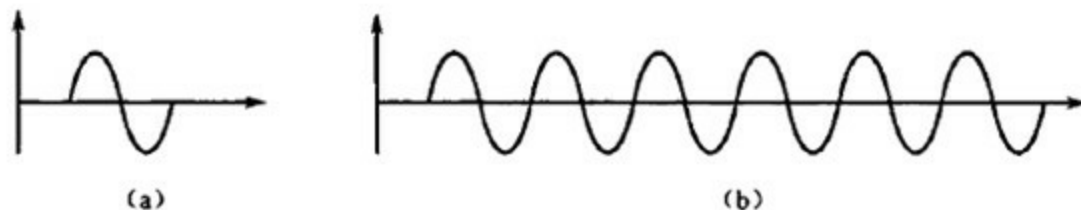


图4.11 交流电的图像

注意，我们扩展了坐标系的Y轴，以显示两种不同的电压极性。很明显，这是两个平滑的圆弧，而不是两个三角形。因为导体做的是圆周运动，它的旋转路线就是圆弧，所以产生的电压当然也是圆弧

为了持续地产生电流，导体需要不停地旋转，所以它的图像也在不断重复，因此图4.11(b)就是交流电的图像，直到导体停止旋转。

离题太远了，原来要说的是爱迪生和特斯拉之间的纠纷，那时爱迪生已经上了年纪，加上他的权威地位，变得非常固执，当特斯拉劝他使用交流电时，他非常反感，再说他已经在直流电上投入了大量的金钱和精力，他要保护已有的投资。

这可能还不是他们最后分道扬镳的最直接原因，据说有一次他们在一起讨论有关发电机革新的问题，爱迪生对特斯拉说如果他能取得成功，他会支付给特斯拉5万美元

好的消息是特斯拉取得了成功，坏消息是他没有拿到5万美元，更糟糕的是爱迪生对他说“你不知道我们美国人爱开玩笑吗？”特斯拉认为自己在这件事上受到了侮辱。

在这种情况下，他愤而辞职，投靠了另一家公司，并建立了自己实验室，专心研究交流电传输技术，而他的秘密武器就是变压器。

其实变压器非常简单，拿一个铁框，然后用绝缘体导线在它的两边分别绕上线圈，左边的线圈称为初级线圈，右边的线圈称为次级线圈，如图4.12所示



图4.12 变压器示意图

如果把初级线圈接在交流电上，铁框就变成了一个电磁铁，而且非常特殊的是因为交流电的性质，决定了这个电磁铁的南北极和磁场强弱都在不停地变化着

这么说来变压器好像没什么大用处，事实上它的用处很大，如果初级线圈有1000匝而次级线圈有5000匝，那么在次级就能获得比初级高5倍的电压，这相当于升压；反之，如果初级有5000匝而次级有1000匝，则次级的电压就是初级的1/5，这相当于降压

为了远距离输送电给那些需要灯泡照明的地区，特斯拉所在的公司首先用变压器把交流电的电压升高，比如升到50kV，然后通过高压输电线路送出支，这样电的损耗就会大大降低

这个电压是非常高的，很危险。如果直接提供给灯泡，会在一瞬间烧毁灯泡。不过好在变压器也能把电压降下来，所以在高压输电线路到达城镇和工厂的时候，再用变压器把电压降低，这样就没问题了

从理性上来讲，作为一名科学家，爱迪生当然明白这一切，但他还是要反对交流电，他利用自己的威望和影响力向公众宣称交流电非常危险，为此，他发表文章，印刷一些攻击交流电的小册子，为了增强宣传效果，增加人们对交流电的恐惧心理，他还找了一些无主的猫和狗，在公众面前用交流电将它们电死，在这方面甚至有人传说他电死了一头大象。

除此之外，双方还对当局进行政治游说，以取得政府对各自技术标准的支持。爱迪生的公司曾希望政府将供电的电压限制在几百伏以内。不过，这也没有用。今天，即使是从全世界范围内来看，电网绝大多数还是交流电。因此，可以说交流电最终获得了胜利

## 4.6 无线电通信的开端

最开始，电报和电话都采用电线或电缆来进行远距离通信，但这是一个非常不容易的事业：需要花费大量的金钱，而且很难增加通信距离——导线越长，所具有的电阻越大，这对发送和接收设备都是个考验。

通信的历史说到这里的时候，无线电波的时代到来了！

无线电波是一种电磁波。那么电磁波是怎样产生的呢？

要得到电磁波，最省力的办法是等闪电。不过要想检测电磁波的存在，你还得准备一个收音机。当天空中乌云密布、电闪雷鸣的时候，打开收音机会听到收音机发出“喀喀喀喀”的声音，表明闪电的确发出了电磁波，而且你也收到了它

用这种方式来验证电磁波的存在当然不是好办法。好的消息是天空并不是唯一能够产生电磁波的地方，通常其他方法——有时候甚至是非常容易的方法就能产生电磁波。找一部收音机、一节电池以及一段电线，将收音机调到没有电台的位置，电线的一头与电池的正极相连，然后用电线的另一头在负极上反复划过，如果收音机离得不太远的话，会听到“喀啦，喀啦”的声音，这就是说，这样也能产生电磁波。与闪电相比，这种方法没有任何限制。

第一个预言电磁波存在的人是麦克斯韦，1831年6月出生于英国爱丁堡。1873年，他的著作《电磁学通论》问世，用数学的方法预言了电磁波的存在，并证明它的速度和光速一样。但在当时谁也没有见过电磁波，也无法证明他的预言是否正确，包括他自己。

刚迷上无线电的时候，我懂的东西不多，但对什么都好奇，也愿意动手。那有次一时高兴，决定自己制作一个变压器。在初级线圈上绕了好几千匝，没有图4.13好看，我也没有万用表这样的工具，为了知道这个线圈到底通不通，我把它接到电池上，顺便看看这个电磁铁有多大磁性。就在这个大线圈和电池断开的一刹那，我感觉自己的胳膊肘有种被重重敲击的感觉。我曾经认为这是错觉，因为一个大线圈和普通的电池就能让人感到这么大的电击，实在没有道理

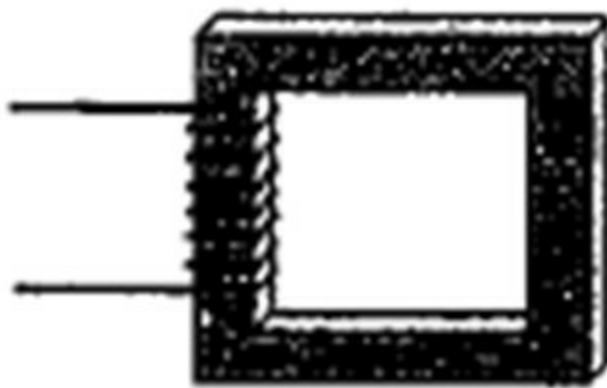


图4.13 带铁框（芯）的线圈

然而这并不是错觉，而且当然是有道理的，只是我不知道。事实上，早在200年前，那位伟大的电磁铁大师亨利就有过类似的发现，当时他正在研究用电磁铁吊起那些巨大的铁块。他发现，当电磁铁断电的一瞬间，在开关上竟然拉起了一道明亮的电弧，也就是说，当电磁铁断电的一瞬间，绕在它上面的线圈就产生了非常高的电压，在电磁学中，这叫做自感。

自感能够产生瞬时高压，它可以发生在线圈断电的一瞬间，也可以发生在线圈通电的一瞬间，可以这样认为：当线圈通电或断电时，它的磁场会剧烈发生变化，从而在自身产生感应电流。通常，影响自感强烈程度的因素包括线圈匝数、形状以及绕在哪种类型的铁芯上

从严格的电磁学来说，产生自感的原因是在开关闭合或断开的瞬间，电流的大小急剧变化，从而产生了一个同样迅速变化的磁场，这个磁场反过来在同一线圈中产生电磁感应，已经说过，自感产生的电压非常高，通常在开关断开时，产生的自感电压可能是它原来电压的几倍甚至是几十倍。这么强的磁场，这么高的电压，它们瞬间产生又瞬间消失，就像什么事情也没有发生一样。

自感或许很有用，也许能用来产生电磁皮，反正，不试一试谁知道呢。首先，如图4.14那样，先来做一个变压器

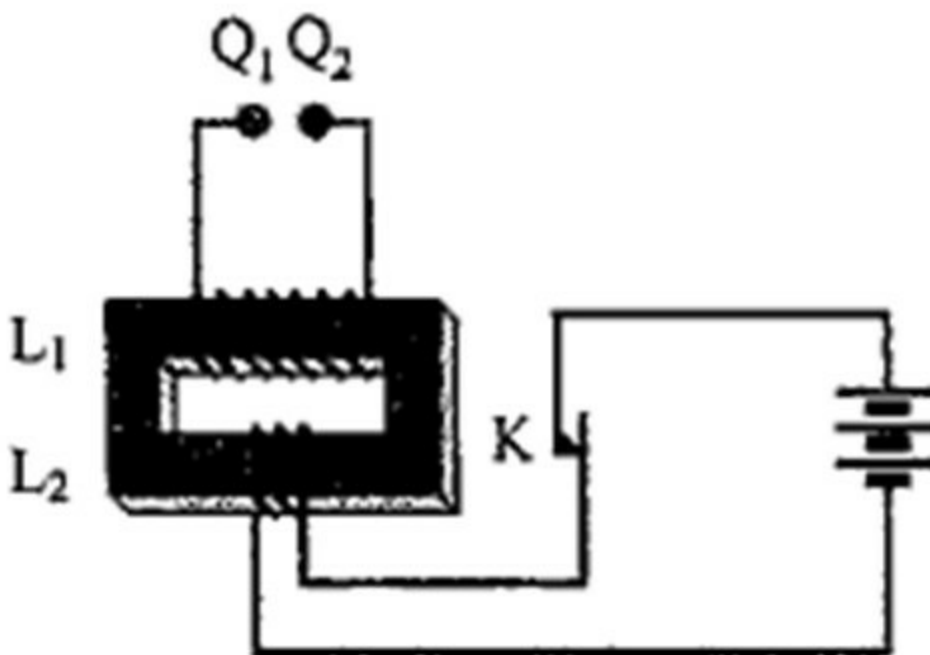


图4.14 利用自感原理制成的火花式电磁波发生器

这个变压器的两个线圈分别是 $L_1$ 和 $L_2$ ，它们的匝数是不一样的。 $L_2$ 通常有几百匝，而 $L_1$ 是它的100-200倍，也就是几万匝

通常，自感发生在开关接通或断开的时候，为了能够持续地产生自感，我们在铁框的旁边安装一个类似于继电器衔铁那样的开关，它与线圈 $L_2$ 以及电源构成一个串联的通路。这样，当整个电路接通的时候，由于K是闭合的，铁框会产生磁性，从而把K吸开，电路断开，电路一断开，K又恢复原状，电路又被接通，就这样“啪嗒，啪嗒”一直不停地进行下去

当自感持续发生的时候，整个铁框中的磁场是一个不停跟着变化、比较强的磁场。在铁框的另一边，线圈 $L_1$ 的两端分别连接着铜球 $Q_1$ 和 $Q_2$ ，由于 $L_1$ 的匝数是 $L_2$ 的几百倍，属于升压变压器，当 $L_2$ 上的自感持续发生时，根据变压器的原理，这会在 $L_1$ 上产生几万伏的高压，使得距离很近的铜球 $Q_1$ 和 $Q_2$ 产生持续的放电。

实际上，这是在模仿闪电。更重要的是，它比闪电容易驾驭。

第一个通过实验证实麦克斯韦预言的赫兹（1857-1894），德国汉堡人，在那个时代，大家对麦克斯韦的预言半信半疑，为了证实电磁波确实是存在的，赫兹制作了一个电磁波发生器，其原理大致与我们前面讲过的相同。

这只是整个问题的一面，电磁波无法用肉眼观察到，需要用其他方法证明它的存在。想知道是不是刮风了，不必到户外去，只要看看外面的树是否在摇摆，赫兹的实验也需要一棵这样的树

赫兹的电磁接收器是用一根粗铜线两头各接一个铜球做成的，如图4.15所示。

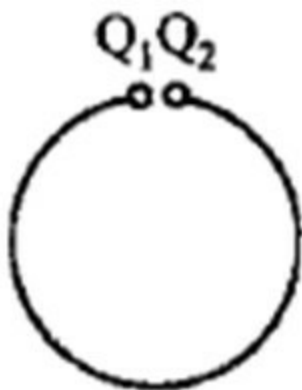


图4.15 赫兹的电磁波接收装置

把铜线弯成圆形，让两个铜球 $Q_1$ 和 $Q_2$ 之间保持一个很小的间隙，当电磁波发生器工作的时候，如果把这个接收器放在不远的地方，并调节两个铜球之间的距离，就可以观察到 $Q_1$ 和 $Q_2$ 之间也会出现微弱的火花。这意味着，那个噼啪作响的东西产生了电磁波，而这个接收器检测到了电磁波的存在。

无线电波是电磁波的一种。无线电波在传播过程中，如果遇到导体都要把它的能量分出一小部分来，在导体内产生电压和电流，所以每个人都是一个人肉发电机，但身体的感应电流非常微弱，除非身处能量非常强大的发射机旁边。但上面的那个例子中，电磁波的成因是什么呢？

在闪电和高压发生器那里，由于高压的存在，有一部分空气被击穿（在高压的作用下，空气分子被迫变成导体，说明这时候空气已经被击穿了）而放电。在这个过程中，参与导电的空气分子越来越多，电流越来越大，而且很不规则，同时，这些空气分子急速升温，并引起空气扰动，这就是火花和响声的来源，之后，电流又逐渐减小，最后火花消失，也没有响声了，电流变为零，整个过程只在一瞬间

这意味着，当电流变化得非常厉害时就会产生电磁波

问题是，电流怎样才算变化得很厉害呢？而且交流电本身就是变化的

从前面的讲解中我们知道，交流电是周期性重复的，导体在磁场中每旋转一周，重复一次。在电学里，重复速率称为频率，单位是赫兹（Hz），为的是纪念赫兹发明的第一个电磁波发生器和接收器。如果1秒重复1次，就称它的频率是1Hz，如果重复100次，就是100Hz

“频率”并不是一个新名词，也不是专门为交流电而创造的名词

在我国，政府对电力供应的各项指标有统一的规定，其中要求交流电的频率必须是50Hz，这意味着，我们平时所用的电，在1秒钟内要经



历50次的正负极翻转和电压起伏。50Hz不算高，但不能再低了，要是小于50Hz，你家里的灯泡就会慢慢亮起来，然后慢慢暗下去，周而复始。

50Hz是一个很低的频率了，就连我们平时说话的声波频率都比它高很多，这样低的频率只能在导线周围产生变化的磁场，根本辐射不出去。要远距离辐射电磁波，除了加大能量外，还需要提高频率，现在，我们用收音机收听无线电广播，频率在500000Hz（500KHz）以上，接收和发送手机信号的频率则在800000000Hz（800MHz）以上，看得出来这样表示很不方便，所以通常使用下面的换算单位：

1kHz=1000Hz（1千Hz）

1MHz=1000kHz=1000000Hz（10万Hz）

1GHz=1000MHz=1000000kHz=1000000000Hz（10亿Hz）

蓝牙技术的频率是2.4GHz

尽管赫兹本人对电磁波的应用前景并不看好，但这并没有影响世界上第一个无线电报的诞生。在后面我们将继续讲述人类通信史的后续发展，不过不要忘了就在赫兹等人忙着让电磁波现形的时候，那些整天为研制自动计算器而殚精竭虑的人们也从亨利、莫尔斯、爱迪生、特斯拉、赫兹这些人那里看到了他们需要的东西。现在是计算机走到前台的时候了

## 第5章 从逻辑学到逻辑电路

全加器的构想表明，我们离这台加法机的完成不远了。那么如何实现全加器的内部构造呢？

这只不过是一台加法机，以后还要制造减法机、乘法机、除法机、平方根机.....这些功能齐全的运算部件

如果能够发明一种方法，通过分析一个电路的输入和输出，然后就能知道这个电路如何构造该多好啊

的确有这样的方法，这种方法和逻辑学还有些关系

## 5.1 逻辑学

思维（大脑的活动）包括形象思维和抽象思维。形象思维是借助于具体的形象，或者外部事物留在大脑中的印象而产生的联想和想象。如果你在野外看到一个蘑菇想到了伞，那么证明你的大脑在形象思维方面是正常的。

形象思维接近于人类的本能，我们一生下来就具备这种潜能。形象思维不拘一格，没有定势，取决于你是否见多识广，也和你个人特点有关，当人们进行艺术创作或发明创造时，用的也是形象思维。比如古代的鲁班，发现带齿的草叶而发明了锯

而且形象思维可以在梦中或者迷迷糊糊时进行，威廉·巴克兰是18世纪著名的地质学专家，有一次，他半夜里突然大喊“天哪！我认为化石上的脚印一定是乌龟的！”然后他和太太来到厨房，摊开面团，把乌龟放在上面观察乌龟踩出来的脚印，果真和化石上的印迹一模一样。

另一些形象思维的例子包括写作、拍电影和绘画等艺术创作过程。写景状物的古诗和猜谜语都和形象思维有关。形象思维比较简单、比较初级，我们每个人很小就具备了这种能力。

抽象思维不借助于头脑中的形象。比如计算 $168+33\times 105$ 或 $100^3$ 这样的数学题。在这里你用不着想象自己能看到168个鸡蛋、33条狗，得出这两道题的结果和你当时所想到的东西没有关系。所以做数学题的过程是抽象思维。

抽象思维无处不在，每时每刻在每个人的脑子里不停地进行着。抽象思维是人类最主要的思维形式，差不多也是人类所特有的。

其他动物可能也有一些简单的抽象思维能力，但不如人类有效。例如，可以训练狗用嘴叼着一桶水去救火，时间久了，这只狗一旦见到着火，就会叼上一桶水去救火。但如果把桶里的水换成汽油，它也会去浇上的。这意味着“水能灭火，油能燃烧，这是油，所以不能用来救火”的抽象思维能力是低级动物所不具有的。

看得出来，和形象思维的想象与联想不同，抽象思维总是要从已知的事实出发计算出一个结果，得到一个新的结论；或用一组被公认为真实的材料，证明某种观点或说法。对于这两个过程，用那些研究抽象思维的人们的行话来说，分别是推理和论证。

抽象思维过程可能掺杂着形象思维，但不是至关重要的因素。比如有一天你看到远处浓烟滚滚，于是你知道远处可能失火了。在这个例子中，有形象思维的存在，比如浓烟，以及你仿佛看到正在着火时的情景，不过这都不是最重要的，最重要的是抽象思维可以把浓烟和着火这

两种看似不想干的事情联系在一起，而形象思维不能。

抽象思维不是凭空进行的，如同形象思维中的形象一样，一定要借助点什么。但和形象思维需要借助的形象不同，抽象思维依赖的是万事万物在大脑中的印象、认识、认知以及对它们有别于其他事物的本质特征的概括。比如前面的油、火、烟等，这些称为概念

要想把概念讲清楚不是一件轻松的事儿，因为你很可能简单地认为它只是从嘴里说出来的，或者写在纸上的一个词语，比如“雨”

不是这么简单，概念事实上存在于你的大脑之中，是你的一种思想活动，是你对万事万物在大脑中的认知。当你看到一棵树，或者听到“树”这个词的时候，你大脑中所反映和意识到的内容差不多就是概念。这里面有大脑中浮现出的形象（有叶子有枝干、根在下，往上方生长……），还包括一些深层次的认知（具有木质实心茎杆的多年生植物）。总体来说，概念是一种思维活动，很特殊，每种事物都有与其他事物区别的特有属性或本质属性，反映在我们的大脑中，就是概念

概念是人们头脑中的思想，即看不见，也听不到，它和语言之间的关系仅仅在于，你必须依靠后者才能表达出来，让别人看得见，听得到。但，即使你认识那些字词，也不表示你头脑中有和这些字词对应的概念。比如“饱和分”、“芳香分”、“极性化合物”、“沥青质”……大部分人不知道这都是些什么，因为虽然认识这些字，但这些字代表了什么在你的大脑中没有概念

除此之外，同一个概念可能对应好多种语言或表达方式，比如“雨”在英语中是rain

对于抽象思维来说，概念仅仅是最基本的元素，是出发点，要想进行真正的思考或推理，概念之间要用连接词串在一起，在大脑中形成一个意思，一个论断或断定，这叫做命题，例如：

3乘以3等于9；

三条边都相等的三角形是等边三角形；

大白菜掉价了；

伞可以遮阳；

如果乘坐公交车时，如果不知道车开往哪里，想知道自己是否应该搭乘这辆车，询问司机，碰巧他心情不好

“这车到哪儿”

“到终点站！”

虽然没什么用，但司机的回答也是命题

多数命题是位于大脑中的经验和知识，这些经验和知识是你从小通过玩耍、推理和学习形成的。同时在这个过程中，你肯定也学会了将概念变成命题的技巧。

从概念到命题、再到推理，这是一个完整的抽象思维的全部过程。换句话说，任何时候，人们要进行抽象思维，必然要依赖于这三种形式。

形象思维不存在正确与否的问题，因为形象思维是发散的、自由的、无拘无束的，没有定势，不需要规则，只要有足够的想象力，想到什么都可以

和形象思维不同，抽象思维通常被认为是在追求真理，因而会出现问题。由于抽象思维包括概念、命题和推理，所以如果这三个中的任何一个出现问题，麻烦就来了。

这不是一个最近才被发现的问题。公元前6世纪的印度、公元前5世纪的中国（战国时期）和公元前4世纪与公元前1世纪的希腊都有人注意到了这个问题。

在我国，最早研究这方面问题的人生活在春秋战国时代，大家可能听说过公孙龙的“白马非马”。他说“马”是就形体而论的，而“白马”指马的颜色，形体和颜色不是一回事儿，所以“白马”不是马

很显然，“白马非马”玩的是概念。但除了这段繁荣的时期之外，由于各种原因，我国在这方面的研究一直没有太大进展，所以我们将目光转向西方。

在西方，这方面的研究起源于古希腊，而且开始于亚里士多德。他是一个哲学家，柏拉图的学生。那时，他的研究是作为哲学的一部分来进行的，因为哲学研究的是精神领域的问题，而思维恰恰与此密切相关。公元前336年，他在雅典开设了吕克昂学园。在教学活动中，亚里士多德通常是在学生的簇拥下沿竞技场的游廊边散步边讲授学问，所以后人一般称这里形成的学派为“逍遥学派”。亚里士多德一生留下了大量著作，其中包括著名的《工具论》

《工具论》不是一本独立的著作，而是《范畴篇》、《解释篇》、《分析前篇》、《分析后篇》、《论辩篇》和《辨谬篇》的总称。在这些著作里，亚里士多德提出了有名的“三段论”，它可以这样表述：

人都是要死的；

苏格拉底是人，

所以苏格拉底是要死的。

这段话流传范围很广。从抽象思维的角度来看，这是一个典型的从概念出发，根据已有的命题得出一个新命题的推理过程

类似于中国的“名”或者“名辩”这样的叫法，在古希腊，亚里士多德研究的对象有“词语”、“言语”、“思维”、“推理”的意思，在英语里对应的单词是logic，在中国近代史上，严复将其称为“名学”，1902年，他翻译引进了《穆勒名学》，后来，干脆把这个外来词语连同它的外国发音

一起引进来，直接叫做“逻辑”，而这门学问则叫做“逻辑学”

逻辑学最早是哲学的一个分支，而哲学则以深奥晦涩著称，因此逻辑学也并不比哲学更容易让人明白，在生活中，有关哲学和逻辑学的著作很少有普通人愿意读，好像研究这门学问的人不愿意让普通人明白他们的思想和成果。哲学语言有时也用于解释某些三言两语说不清楚的事实和理论，例如“地震是地球内部矛盾运动的结果及其外部表现”。看到这句话似乎明白，又模模糊糊，只能认为这句话给我们提供了一个巨大的想象空间

逻辑学是一门实用性很强的科学，现在在联合国教科文组织的学科分类目录中，逻辑学是与数学、物理学等并列的七大基础学科之一

逻辑学的产生和发展说明了一个基本事实，那就是人在抽象思维方面不是完美的，或者说经常是有缺陷的，据说有位美国参议员对逻辑学家贝尔克说“所有的共产党人都反对我，你也反对我，所以你是共产党人”。贝尔克当即回答说“亲爱的参议员先生，您的推论真是妙极了！如果您的推论能够成立，那么下面的推论也能成立：所有的鹅都吃白菜，您也吃白菜，所以您是鹅。”

逻辑学是一门学问，逻辑学的任务就是总结抽象思维的规律和特点，让我们在掌握它之后可以明辨是非、去伪存真。更重要的是让我们在说话和思考问题的时候，从一开始就具有很强的思维能力和很高的思维品质

这可不是唱高调，大家可能都知道，亚里士多德曾经说过，重的物体比轻的物体下落快。但伽利略做了一个思想实验，仅仅用一个抽象思维就推翻了它。伽利略想如果亚里士多德是正确的，那么将一块大的石头和一块小石头绑在一起形成一个更大的石头，当它们下落时，大的石头会被小的石头拖累而总体上变慢；另一方面，因为当两块石头绑在一起后，会变得更重，从而总体上下落的更快。从同一个前提出发，居然得出两个截然相反的结论，只能说明亚里士多德的结论是错的。

逻辑学首先研究概念和命题，并在此基础上形成了一些公认的准则，比如对于亚里士多德的三段论，如果不正确使用，就有可能发生逻辑错误：

鲁迅的作品不是一天能够读完的。

《孔乙己》是鲁迅的作品，

所以《孔乙己》不是一天就能读完的。

以上推理过程中的错误是如此明显，但其中的原因却很隐蔽，不容易看出来，这是怎么回事儿？

从逻辑学中，概念及其运用是一个复杂的话题，不是想象中那么简单。在亚里士多德那里，“人都是要死的”和“亚里士多德是人”这两句话

里面的“人”是同一个概念。而在“鲁迅的作品不是一天就能读完的”和“《孔乙己》是鲁迅的作品”这两句话中，尽管都有“鲁迅的作品”但不是同一个概念，前者是鲁迅作品的总称，而后者特指《孔乙己》，这样就违反了三段论的格式，以至于发生了错误。逻辑学要求，在一个单独的抽象思维过程中，概念和命题必须保持一致，这叫同一律。如果违反了同一律，就会发生我们日常生活中所说的“偷换概念”、“偷换命题”、“混淆概念”这类错误

多数情况下，我们都在遵守同一律，只是自己没有意识到。拥有这种技能，是在我们出生后一直从周围学习并不断强化的结果。不过在这个过程中，我们同时也学会了偷换概念或转移话题的本领。

同一律不是唯一的逻辑准则，其他的还有矛盾律、排中律等等。特别是矛盾律，我们应该最熟悉，其实就是“自相矛盾”。从逻辑学上来说，在一个独立的抽象思维过程中，互相对立的命题之间不能同时为真，也不能同时为假，这就叫矛盾律

亚里士多德开创的逻辑学很古老，超过了两千年。所以相对于逻辑学后来的发展而言，它是古典逻辑。

古典逻辑研究最基本的逻辑准则和逻辑规律，当然也研究推理形式，但人的思维是复杂多样的，所以逻辑推理也有各种各样的形式。比如，在过去的三年里，某学校在全省会考中都取得了第一名，这样人们会倾向于认为，明年的全省会考这个学校同样会取得第一名。可以看出，这是基于归纳以往的情况而得出的结论，叫归纳推理，但归纳推理得出的结论是靠不住的，有可能是真，也有可能是假的，错的。谁能证明明年的会考中这个学校一定会得第一呢？

再比如，老张得了感冒，他想到，前段时间老刘也得了感冒，而且自己的症状和老刘一样，于是他觉得，问问老刘吃的什么药，自己也吃感冒就好了，这叫类比推理，但和归纳推理一样，类比推理同样不能确保一定是正确的

也许对于亚里士多德来说，像归纳、类比这样的推理形式实在是没有办法来保证它们总是正确的，但他发现了三段论

三段论有很多种形式，其中最典型的一种格式是这样的：

所以M是P；

所以R是M；

所以，所有R是P。

一旦固定了这样的格式，在任何时候，如果大脑中出现了这样的抽象思维，只要前两句是真命题，那么推理的结果也就是第三句话，就必然也是真的。唯一需要注意的是，在前两个命题中，M必须遵守同一律，也就是保持同一种概念不变。

看起来亚里士多德是在发明一种公式，或者说一种形式。要想得出一个真的、正确的结论，只需要严格套用这种公式，并确保前两个命题为真就行了。这也使得他开创的逻辑学在后来的岁月中有了另外两种名称：演绎（从前提必然得出结论的推理，从一些假设的命题出发，运用逻辑的规则导出另一命题的过程）逻辑和形式逻辑。

演绎逻辑或者说形式逻辑，侧重于从形式上进行推理和论证，也就是“演绎”。尽管像归纳这样的推理形式在生活中可能用得更多，但它们却不是主要的研究对象。这样，在这门学问中，形式就成了追求真理的主要手段，除了上面所说的三段论外，其他的逻辑形式也很多，比如联言推理、选言推理等等。

要想了解联言推理，举一个例子可能是最好的方法。这里有一道推理题，可以用来训练小学生的思维能力：在桌子上有两张牌，2的右边是5，红桃的左边是方块，请问这是两张什么牌？

这当然是一道非常简单的题，不过重要的不是题目本身，而在于解决这个问题的思维过程。看完题目后，很自然地意识到这是一个命题：

左边的牌是2；

右边的牌是5；

左边的牌是方块；

右边的牌是红桃。

现在我们的大脑会进行联言推理，从这些命题得出结论，首先，把这些较小的命题组织起来，形成一个个更大的命题：

左边的牌是2，而且左边的牌是方块。

右边的牌是5，而且右边的牌是红桃。

两个小命题“左边的牌是2”和“左边的牌是方块”结合在一起，形成一个更大的命题，这称为联言命题。而它的每一个前提“左边的牌是2”、“左边的牌是方块”称为联言命题的支命题，简称联言支。

联言推理的第一种形式是组合，就是从支命题推出一个结论。比如，从

左边的牌是2，而且左边的牌是方块

可以推出

左边的牌既是2又是方块（也就是说左边的牌是方块2）

从上面的联言推理过程可以看出，如果所有的支命题都是真的，则推理结论就是真的；而只要有一个支命题为假，则推理结论就是假的。要是“左边的牌是2”这个命题为假，那么结论“左边的牌是方块2”就不可能是真的。这样，只要判断一下所有支命题的真假就可知道推理结果的真假

这是联言命题的正向推理过程，反过来，如果推理结论是真的，则可以断定所有的支命题都是真的。但，如果推理结论是假的，则无法判



断哪一个支命题是假的

比如要是事实证明左边的牌不是方块2，不能说左边的牌不是2，因为它可能是2，但不是方块，也可能即不是2也不是方块，也可能是方块，但不是2，反正都有可能。了解到联言命题这一特点，有助于我们在生活中提高正确推理和辨别错误推理的能力

选言推理和联言推理一样，选言推理也是在两个或两个以上的命题之间进行，但，支命题之间的关系是松散的，举个例子：

小张是马老师的学生或者是刘老师的学生

在联言推理中，支命题之间的连词通常是“并且”，在选言推理中，支命题之间的连接词通常是“或者”，有一种选择的意思。这些支命题共同构成了一个更大的命题，这就是选言命题

“小张是马老师的学生，或者是刘老师的学生”这个选言命题有可能是假的，因为小张也许既不是马老师的学生也不是刘老师的学生。这是一种特殊的情况。选言推理研究的对象是那些支命题中至少有一个为真的选言命题

在这种情况下，逻辑学对于选言命题的第一个重要的结论是：如果一个支命题为假，那么其他支命题至少有一个为真。比如：

小张是马老师的学生，或者是刘老师的学生。

小张不是马老师的学生。

所以小张是刘老师的学生

这个推理的过程是严密的，正确的，没有问题。但如果已知一个支命题为真，是否就能判断出其他命题都为假呢？像这样：

小张是马老师的学生或者是刘老师的学生

小张是马老师的学生

所以小张不是刘老师的学生

上面的推理是有问题的，因为尽管小张是马老师的学生，但不影响小张也可能是刘老师的学生，所以有关选言命题的另一个重要结论就是：已知一部分支命题为真，不能推出另一部分支命题的真假

张三的庄稼让羊啃了，据说是王五的羊。他去找王五赔偿所有损失。王五说“我家的羊啃了你的庄稼，这我承认，我也赔。但这村子里养羊的不止我一家，凭什么肯定全是我的羊啃的？”

这是一个有关选言推理的小故事，其核心就是一个错误的推理过程：

地里的庄稼是李四的羊啃的，或者是王五家的羊啃的。

是王五家的羊啃的。

所以不是李四家的羊啃的。

错误的原因在于这一类的选言推理中，所有的支命题之间不是排斥

和对抗的关系，它们有可能同时为真。像这样的选言推理称为相容的选言推理

这意味着，还应该有一种选言推理形式，即不相容选言推理。的确是这样，在这种推理形式中，所有的支命题只能有一个为真，不能同时为真，比如：

小张要么是男的，要么是女的。

那个人要么来自法国，要么来自德国，再不就是来自瑞典

对于不相容的选言推理，也有一套规则。事实上，除了三段论、联言推理、选言推理之外，形式逻辑还有其他大量的推理形式，而且有时它们还组合起来形成复杂的推理形式

逻辑学的任务就是寻找获得真理的方法。

以比喻代替推理和论证是坏习惯

形式逻辑的发展已经有两千多年了，在这期间出现了许多杰出的逻辑学家，他们以及他们的追随者从来没有停止过争吵 – 从什么是逻辑学到逻辑学到底应该包括哪些内容，等等。这也直接导致了其他逻辑学门类的产生，比如数理逻辑、多值逻辑、直觉逻辑、亚结构逻辑、模态逻辑、辩证逻辑等等。也许只有一点大家都认同的就是尽管形式逻辑不是万能钥匙，解决不了所有的逻辑学问题，但不可否认的是它应当永远在整个逻辑学中占有一席之地

## 5.2 数理逻辑

总的来说，数学是一种强有力的工具，几乎所有的学科要达到完善的地步，都应当能用数学进行完美的描述。而对于数学本身来说，简洁的、能够恰当地描述各种事物内在本质的符号至关重要。

在历史上，各门学科之间的交叉与融合是一种常态。到17世纪，由于数学的大发展，使得那些既精通数学，又对其他学科有深入研究的人开始想入非非。其中有两个人很特别。

第一个是德国的莱布尼茨，他有一些奇特的想法，觉得人类需要一种普遍的、恰当的符号，普世的所有问题和思想都可以归结为这些符号，然后用一套计算方法来代替人类的思考和推理过程。而且，他希望人类能够发明这样一种机器，能够自动地代替大脑的逻辑思考过程，不管谁和谁有什么样的问题，发生了什么样的争端，只要把相关的前提条件输入这台机器，就能通过“计算”方式，解决问题

莱布尼茨终究也没有发明“普遍符号”，更不要说那样的机器了。一百多年后，又出现了一个人，把逻辑学和数学结合起来创立了数理逻辑，这个人是乔治·布尔

乔治·布尔1815年生于英格兰，他的父亲是一位鞋匠，母亲曾是女仆。12岁时，布尔就掌握了拉丁语和希腊语，后来又自学了意大利语和法语，16岁开始任教以维持生活。

两千年来，亚里士多德一直都是权威，莱布尼茨活着的时候，他有改进传统逻辑的想法，但心存顾虑，也可能是力不从心。到布尔决定要做些什么。

对于传统的形式逻辑来说，三段论一直是金字招牌，无论布尔想怎样改进这门学科，都必须先把它拿下。为此，布尔在前人的基础上，使用集合这个数学工具来研究三段论

所谓集合就是具有共同特性的事物的总体。比如宇宙中所有东西可以看成是一个集合；动物园中所有动物可以形成一个集合；动物园中的所有斑马可以形成一个集合……

挑出两个集合中都有的东西来形成一个新的集合，称为两个集合的交集。比如，如果一个集合中有芝麻和绿豆，另一个集合中有绿豆和苹果，那么这两个集合的交集就是一个新的集合，只包含绿豆。

除了交集外，也可以做相反的事情，那就是把两个集合掺和在一起，形成一个更大的集合，称为“合集”或“并集”。比如前面那两个集合的并集也是一个新的集合，包含绿豆、苹果和芝麻。

一直以来，逻辑学中的概念和命题都是通过自然语言来表达的，论

证或推理的结果也是用自然语言说出来或写成文字。但布尔把它们变成了字母和符号，比如，用M代表人的集合，这里面包含了人全体人类，用P来代表所有会死亡的东西，同时，他还借用了数学里面的一些运算符，如+，×表示概念和命题之间的逻辑关系，×表示两个集合相交，+表示合并两个集合，这样，“人都是会死的”就可以表示成：

$$M \times P = M$$

这个算式要表达的意思是“人”和“会死的东西”的交集是“人”

有了这样的经验，同样可以把“苏格拉底”看成集合S，而“苏格拉底是人”可以表述成

$$S \times M = S$$

意思是“所有的人”和“苏格拉底”的交集只能是“苏格拉底”自己。

因为 $M \times P = M$ ，所以我们可以将M代入 $S \times M = S$ 中，得：

$$S \times M \times P = S$$

因为有 $S \times M = S$ ，代入 $S \times M \times P = S$ ，得：

$$S \times P = S$$

表明会死亡的人和苏格拉底的交集是苏格拉底自身，从而证明了苏格拉底也是会死的。

可以看出，布尔的工作主要是对逻辑进行数学化，并成功地创立了一门新的学科 – 逻辑代数。有时也被称为布尔代数。用布尔代数解决逻辑问题还有一个显著的好处，那就是同一个证明过程可以用来解决不同的、但本质上属于同一种类型的逻辑问题。比如下面的证明过程同样适用于三段论：

金属可以导电。

铅是金属。

所以铅可以导电。

字符以及×和+运算符也可以用在其他逻辑形式上，比如联言推理和选言推理，也就是命题演算（按一定的原理和公式对命题进行计算）

在布尔代数里，可以用字母表示一个命题，比如，用A表示命题“左边的牌是2”，用B表示“左边的牌是方块”。因为在传统的形式逻辑中，一个命题不是真的就是假的，没有其他可能，所以用0代表假，1代表真，这样，命题A和B就只能有两个可能的值0或1。如果A命题为真，则

$$A=1$$

否则

$$A=0$$

除此之外，不可能再有其他值，如果 $A=3$ ，这在逻辑上没有任何实际意义。

在联言推理中，各个支命题之间是并列关系，通常用“并且”来连

接，为了表示这种逻辑关系，布尔代数使用 $\times$ 这个符号，这样，一个联言命题可表示成：

$$A \times B$$

有时，为了方便把它写成 $A \cdot B$ ，或者干脆害怕成

$$AB$$

这样，如果各个支命题都为假，则联言推理的结果就是假的：

$$A \times B = 0 \times 1 = 1 \times 0 = 0$$

只有在所有的支命题都为真的情况下，联言推理的结果才为真：

$$A \times B = 1 \times 1 = 1$$

反过来，如果已经知道支命题 $A=1$ ，联言推理的结果为假，即

$$A \times B = 1 \times B = 0$$

则很容易推理（计算）出支命题 $B$ 为假，即 $B=0$

每个命题都有真假，要么是1，要么是0，这叫真值，也叫逻辑值。“真值”的意思是它本来的值，真正的值。因为它到底是真是假，不以你的看法为转移，有时，它本来是假的，但你却误以为它是真的

任何一个联言命题，它的真假与其各个支命题之间的关系如表5.1所示（假如只有两个支命题），这叫做真值表

表5.1 联言命题的真假表

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

显然，只有各支命题都为真时，联言命题才为真

任何一个命题，比如命题 $A$ ，不管是真是假，它的对立面是“非 $A$ ”，可以表示成 $1-A$ 或 $\overline{A}$ 。显然，如果 $A=1$ ，则 $\overline{A}=0$ ；如果 $A=0$ ，那么 $\overline{A}=1$ ，它们总是相反的。在同一个抽象思维过程中，必须保证概念和命题的前后一致性，否则就会违背矛盾律，以致于说出来话自相矛盾。矛盾律可以表示成：

$$A \times \overline{A} = 0$$

与联言命题不同，在选言命题中，各支命题之间用“或者”、“要么”来连接，是一种选择关系，为了表示这种命题关系，布尔代数用 $+$ 这个符号。比如对于选言命题

小张是马老师的学生或者是刘老师的学生  
来说，如果用 $A$ 代表命题“小张是马老师的学生”，用 $B$ 代表命题“小张是刘老师的学生”，那么上面的选言命题可表示成：

$$A+B$$

因为相容的选言推理必须在整个选言命题为真的前提下进行，这意味着：

$$A+B=1$$

这样，如果已知 $A=0$ ，即“小张是马老师的学生”为假，则必然可推出 $B=1$ ，也就是说“小张是刘老师的学生”为真，因为：

$$A+B=0+1=1$$

但如果 $A=1$ ，即“小张是马老师的学生”为真，则不能断定 $B$ 是0还是1，也就是说“小张是刘老师的学生”这句话不知道是真还是假。因为：

$$A+B=1+0=1$$

$$A+B=1+1=1$$

到目前为止，所有的逻辑运算看上去都和数学里的乘法和加法一样，没有什么区别，但在这里， $1+1$ 明显违背了数学规则

但这毕竟是逻辑学不是真正的数学运算，而 $+$ 和 $\times$ 也都不是这个符号的原来意义。因此，不管一个命题 $A$ 到底是真还是假，谎言重复一千次还是谎言，真理重复一万次也不会发生改变：

$$A+A+\dots+A=A$$

对于 $\times$ 也是这样：

$$A\times A\times\dots\times A=A$$

这其实是同一律在布尔代数中的表示方法，也就是说 $A+A=A$ 及 $A\times A=A$ 实际上表示在同一个抽象思维过程中，概念和命题是要保持不变的。

选言命题的真假与其支命题之间的关系同样可以体现在真值表中，如表5.2

表5.2 选言命题的真值表

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

注意，表中背景颜色较深的那些行，在相容选言推理中，不允许同时出现所有支命题都是0的情况；在不相容的选言推理中，所有的支命题既不允许都为0，也不允许都为1，所以，这张表是两种选言推理形式的结合体。

同几乎所有新生事物一样，布尔的研究成果一开始受到的并不是好评。欧洲大陆的数学家甚至轻蔑地认为它毫无数学意义。然而，布尔的贡献是不可能被一直埋没的，人们很快认识到了它的重要性。

布尔发明了逻辑代数，以此为基础，他及他的支持者们最终完整地建立了一门新的学科 – 数理逻辑，或者叫“符号逻辑”。不幸的是，现在这门学问已经不像是逻辑学更像是数学或者物理学。

至于布尔尽管从逻辑学角度来看“人得肺炎是要死的”这个命题并不一定成立，但肺炎却要了他的命，1864年12月8日，他感染了肺炎，并不幸去世，终年59岁。

## 5.3 数字逻辑和逻辑电路

布尔的理论不仅只受到逻辑学者的敬仰，伟大的理论总会给人以启迪，其中有一个叫克劳德·艾尔伍德·香农（Claude Elwood Shannon）

香农1916年出生于美国密歇根州，从小热爱机械和电器。1936年，香农毕业于密执安大学工程与数学系，在麻省理工大学攻读硕士期间，他选修了布尔的逻辑学

布尔把逻辑学和数学结合起来开创了数理逻辑，香农把布尔代数（逻辑代数）和电学结合起来，开创了一个新的领域：开关电路。

1936年，他20岁时写了一篇论文《继电器和开关电路的符号化分析》，第一次面向大众，系统化地阐述了布尔的逻辑系统和电路通断之间的关系。在布尔代数里， $X$ 代表一个命题， $X=0$ 表示命题为假； $X=1$ 表示命题为真。香农发现如果用 $X$ 代表一个由继电器和普通开关组成的电路，那么 $X=0$ 就表示开关合上； $X=1$ 表示开关打开，如图5.1(a)所示

接着，同样是在论文的第二部分，他进一步阐述了串联电路和并联电路与逻辑学中联言命题及选言命题的一致性。如果直接引用香农论文中的文字和图片会更有说服力，但遗憾的是，香农用0表示电路接通（有电流通过），1表示电路断开（没有电流），而不是现在流行的方法（0代表电路不通，1代表电路接通），所以我们按现在的方式来介绍香农的成果。

联言命题演算相当于两个开关 $X$ 和 $Y$ 的串联，如图5.1(b)所示，只有当两个开关都接通时，整个电路才是通的；两个都断开，或者开关中的任一个断开，整个电路就是断开的。

选言命题演算相当于两个开关的并联，如图5.1(c)所示，两个开关只要有任何一个接通，或者两个同时接通，整个电路就被接通，只有两个开关同时断开，整个电路才是断开的

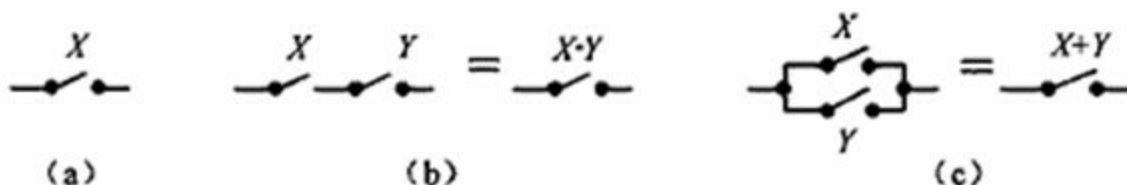


图5.1 命题演算和现实的开关组合具有完美的一致性

按这种观点，布尔代数公式也有了新的解释，见表5.3

表5.3 布尔代数与开关电路的对应关系

布尔代数：对应的开关电路

0·0=0：一个断开的开关和另一个断开的开关串联，整个电路是断开的  
0+0=0：一个断开的开关和另一个断开的开关并联，整个电路是断开的  
1·1=1：一个闭合的开关和另一个闭合的开关串联，整个电路是连通的  
1+1=1：一个闭合的开关和另一个闭合的开关并联，整个电路是连通的  
1·0=0+1=1：一个闭合的开关和另一个断开的开关无论以什么顺序并联，整个电路都是连通的  
1·0=0·1=0：一个闭合的开关和另一个断开的开关无论以什么顺序串联，整个电路都是断开的



事实上，不管由开关组成的电路多么复杂，布尔代数一样可以很好地对其进行解释。比如这样一个电路（图5.2）

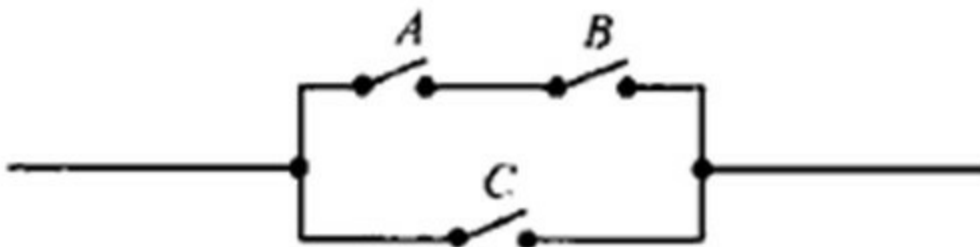


图5.2 布尔代数可以用来解释复杂的开关电路

在这个电路中，开关A和B串联，具有逻辑乘的关系，即 $A \cdot B$ ，或者写成 $AB$ ；同时， $AB$ 和C又是并联的，属于逻辑加的关系，据此可以很容易地得到与整个电路等效的逻辑表达式：

$$A \cdot B + C$$

要是电路特别复杂，开关非常多，那么想知道闭合或打开某个开关会对整个电路有什么影响，这可能是非常麻烦的事情。幸好现在有了布尔代数，使用它会给我们的工作带来极大的方便。比如，在图5.2所示的电路中，所有的开关都断开整个电路还是连通的吗？

这只需“计算”一下就行。因为所有的开关都断开，所以运用逻辑代数的基本规则可知：

$$A \cdot B + C = 0 \cdot 0 + 0 = 0$$

结果为0表明整个电路是断开的。再比如，要是开关C闭合，整个电路还是连通的吗？再算一下：

$$A \cdot B + C = 0 \cdot 0 + 1 = 1$$

计算的结果1表明整个电路是连通的。结合图5.2，这与实际情况一致

真值表是非常有用的工具，简明直观。因为开关的通断和逻辑的真假有着对应关系，都可以方便地用0和1表示，所以一个开关电路的状态也可以通过真值表直观地描述。在这个例子中，用0表示开关断开，1表示开关接通，那么三个开关无论断开还是接通，共有8种组合，在每一种情况下它们与整个电路的状态如表5.4所示：

表5.4 三个开关的状态与电路通断之间的关系

A	B	C	$A \cdot B + C$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1

1 1 0 1  
1 1 1 1

非但如此，香农发现，所有的布尔代数基本规则都非常完美地适用于继电器和开关的电路。比如，假如 $x, y, z$ 都是一些开关，那么：

$$x+y=y+x$$

$$xy=yx$$

$$x+(y+z)=(x+y)+z$$

$$x(yz)=(xy)z$$

上面这些都是显而易见的。另外，要是把0看成一个始终断开的开关，把1看成一个始终闭合的开关，那么对于开关 $x$ 来说，下面的表达式也是成立的：

$$0+x=x$$

$$0 \cdot x=0$$

$$1+x=1$$

$$1 \cdot x=x$$

香农不是第一个发现布尔代数和开关电路之间具有相似性的人。1935年，前苏联莫斯科州立大学的谢斯塔夫也有类似的理论，但直到1941年才首次公开，而香农是在1938年提出的，比谢斯塔夫早一些，影响了后人对他们的评价

这样研究开关电路有用吗？要想让一个电路接通或断开只需要一个开关就够了，这本是件极简单的事情，干吗要把一个电路弄的那么复杂，设置那么多开关？把开关扳来扳去难道不嫌麻烦？研究这些到底有什么意义？

开关电路非常有用，甚至改变了20世纪后半部分的历史，但靠的不是用手来控制里面的开关

传统上开关就是开关，它可以嵌在墙上或者固定在其他某个地方用以控制电流的通断。这些开关有一个共同的特点 – 要打开或关闭它们，只能手动。

但人们发明各种机器的目的并非要操劳于其中，而是要实现自动化，把自己解放出来。所以要是我们用电压和电流来代替人手去控制一些开关，就一样能改变电路的通断状态，而且可能会更有用，比如下面的这个例子（见图5.3）

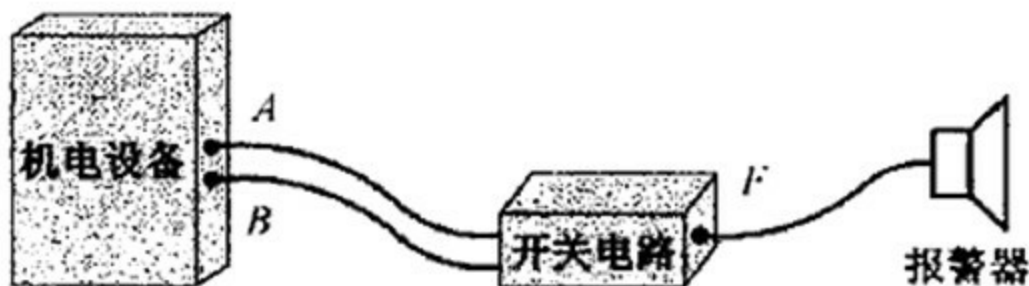


图5.3 用开关电路组成一个报警系统

在这个例子中，有一台大型的机电设备，需要对它的工作情况进行监控，以保证在出现故障时尽快修复。监控的方法是不停地测量这台设备上A,B两点的电压

由于设备内部构造的关系，如果A,B两点都没有电或都有电，表明设备是正常的；要是有一个有电一个没电，那就表明发生了故障。

所以现在需要依据这台设备的特点，设计一个新型的开关电路，可以根据A,B的情况来控制另外一条线路F的通断，如果A,B不正常，开关电路就使F接通，报警器报警，通知故障的发生。

要用电流的通断来控制电路的开关需要使用继电器，我们已经认识了继电器。最简单的电流开关如图5.4所示，仅使用一只继电器，当有适当的电压加在A端时，有电流通过继电器而使它吸合，从而使F端接通

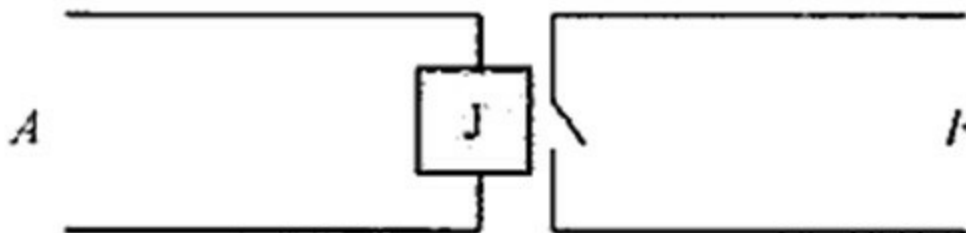


图5.4 继电器的作用是间接地控制另一个电路的通断

看得出，这本来就是一个继电器，相当于一个间接的开关，A端有输入，则F端也接通

继电器仅仅是一个间接开关，它唯一的作用就是使另一条线路接通或断开，就这么简单，用术语来说属于无源器件。但有时希望一个开关能“自行”产生输出，而不是仅仅把一个电路断开或接通。要实现这个目的，就必须为这个开关配备电源（图5.5）

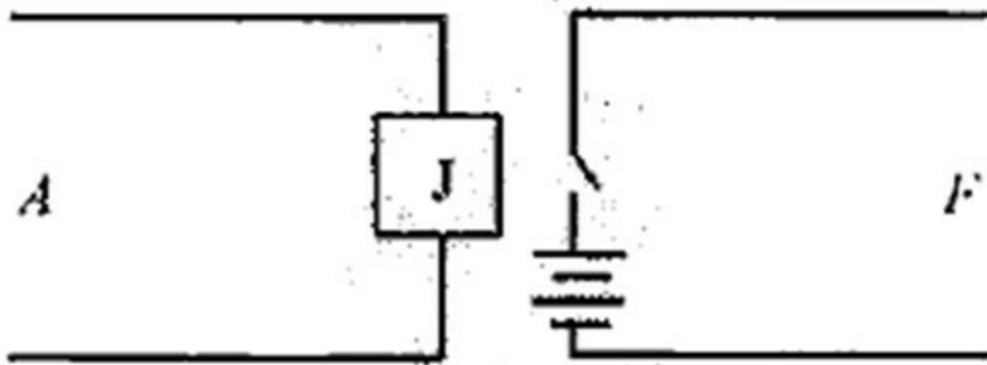


图5.5 一个自带电源的继电器

注意电源现在是这个特殊电路的一部分，这样一来，它就不再是一个单纯的开关，更是在输出什么 – 当然是电能。所以，如果我们把A端看成输入，那么F端就是一个输出，而且F和A之间符合下面的关系：

$$F=A$$

除此之外，利用继电器还可以方便地实现别一种截然相反的功能。如图5.6所示，这次采用的是另一种继电器，平时它处于吸合状态，所以F端可以对外产生输出；当A端加上电压，F端的输出就消失了。所以，在这个电路里，输出F总是与输入A处于相反的状态

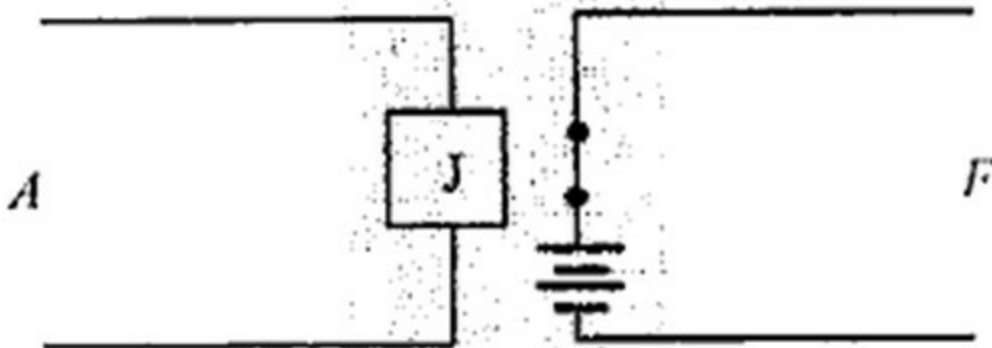


图5.6 一个自带电源的常闭触点继电器属于非门

因为这个原因，它有一个非常专业的称呼 – 非门。

对于非门的应用，一个最简单的例子是用开关为非门提供输入，并用后者的输出控制灯泡的亮灭，如图5.7所示

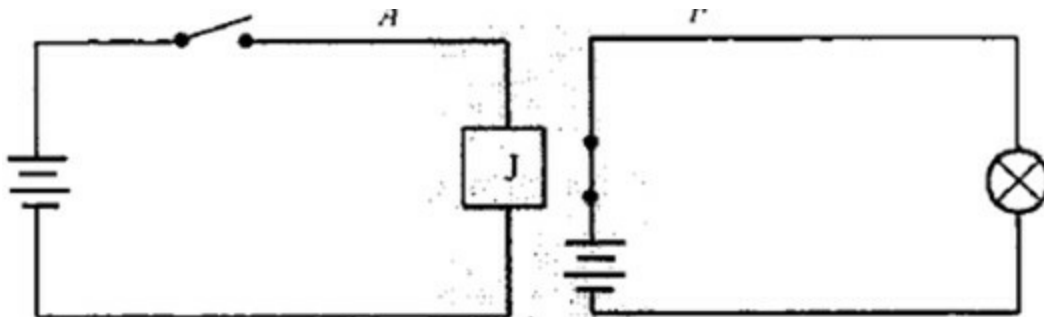


图5.7 一个应用非门的例子

当然这并不是个很好的例子，开灯这样的小事不需要变得这么复杂，此处只是为了说明原理。实际上非门有更多更好的应用，没有它就没有电子计算机

在这个例子中，非门的输入A是由电路左边的开关产生的，而输出F的状态总是和A相反，这可以通过灯泡的亮、灭得到验证

不像手电筒和袖珍收音机这样的家用电器，为门电路装上几节电池是不可思议的做法。原因在后面会知道，门电路被大量地使用。想想看，如果为每个灯泡单独提供一个电源的情形。所以对于任何电器来说，它内部的每个部分都用同一个电源供电，门电路也不例外，在这个例子中，非门的输入和非门本身都使用同一个电源（图5.8）

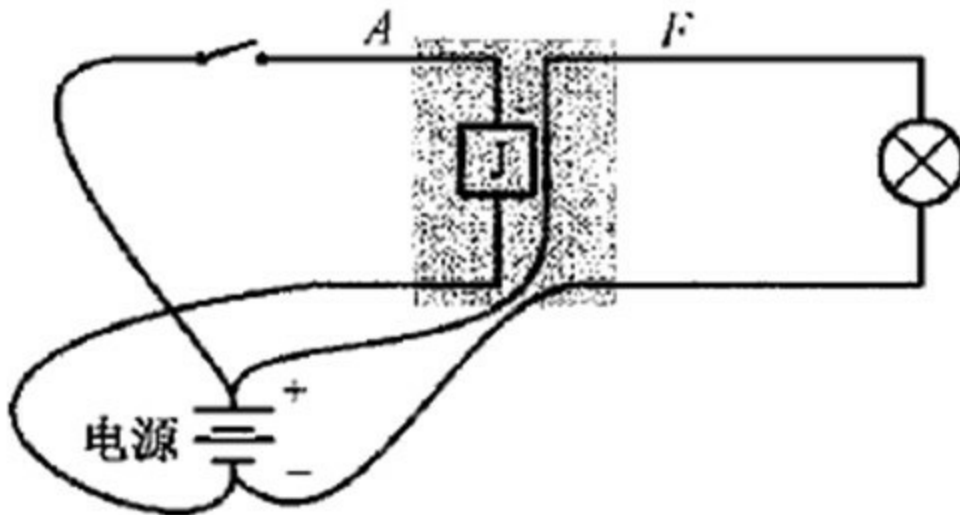


图5.8 在一个完整的电路中，各个组成部分共用电源是通常的做法

这个电路有点古怪，但它能很好的工作。为了看清楚我们在做什么，并尽可能少用一些电线，我们可以借鉴电子工程师的做法（图5.9）

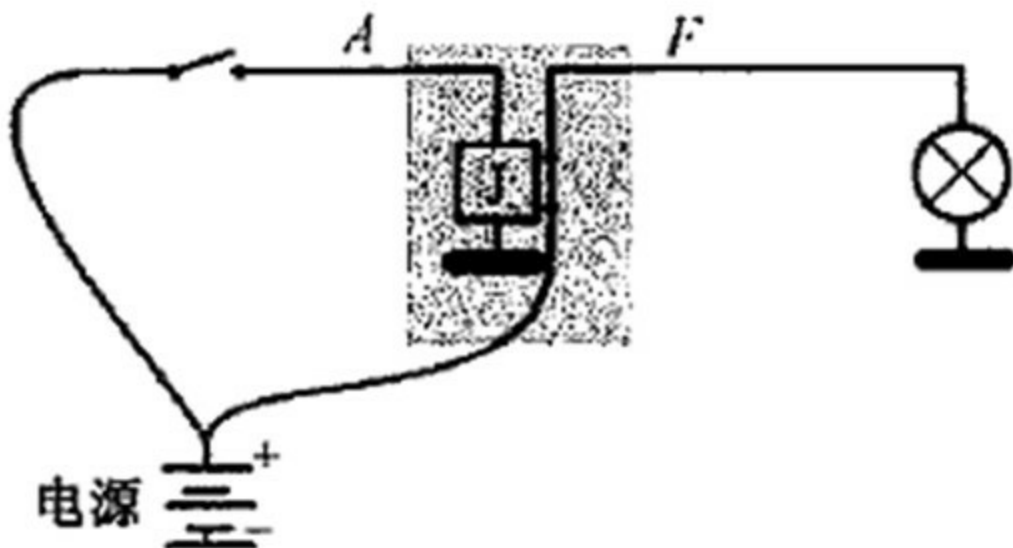


图5.9 为了少画一些连线，应该使用“接地”符号

图中那两个粗短横线的意思是“地线”或“接地”，最早用来表示将电线绑在一根导电的棍子上插到地里，后来也表示电线的交汇点。换句话说，当你按图纸制作电路的时候，必须将所有“地线”接到一起。历史上，地线还有其他几种表示方法，如图5.10所示，不过右边两种已经很少使用了

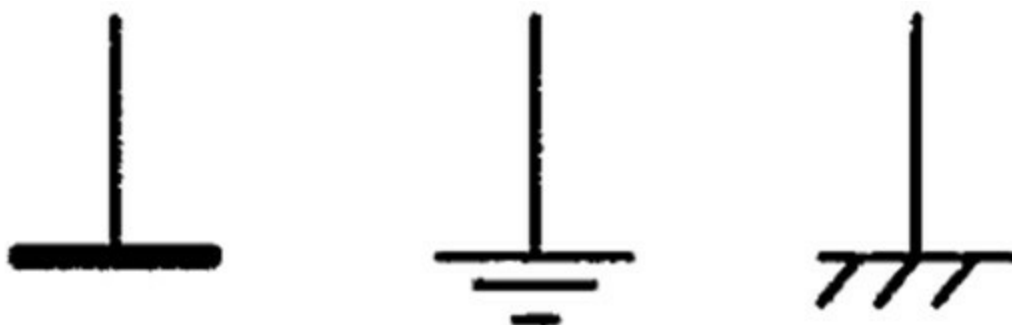


图5.10 曾经使用过的接地符号

尽管图5.9已经很简单明了，但还可以把它变得更简单。通常情况，电源不用画出来的，只用一些符号，比如Vcc是指电源正极（在电子技术领域，通常用字母V来表示电压，cc的意思是circuit，即电路。所以，Vcc通常是指电路供电电压），表示那根电线要接到电源正极，而电路中所有的地线都应当汇集起来连接到电源负极（注意，在这里输入A是由开关提供的，尽管它来自Vcc，当开关闭合时，A等于1；反之则A等于0），如图5.11所示

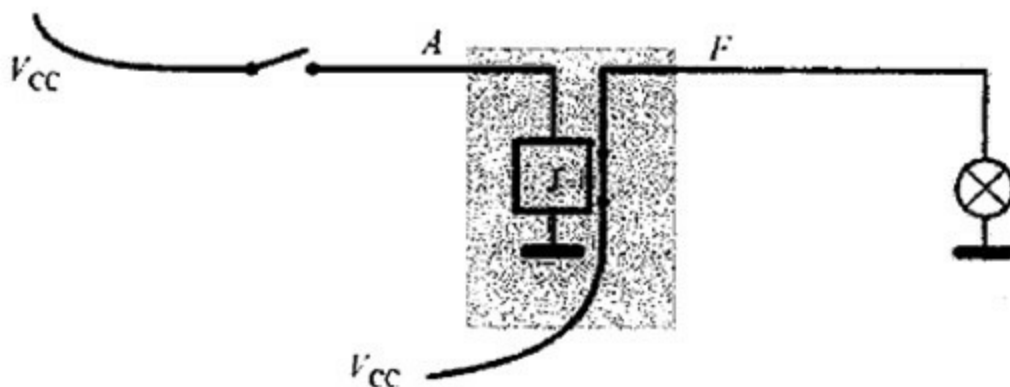


图5.11 在电路图中，电源通常用Vcc和接地来代替

那么，前面一直讨论的非门实际上可以表示成如下的电路形式（图5.12）

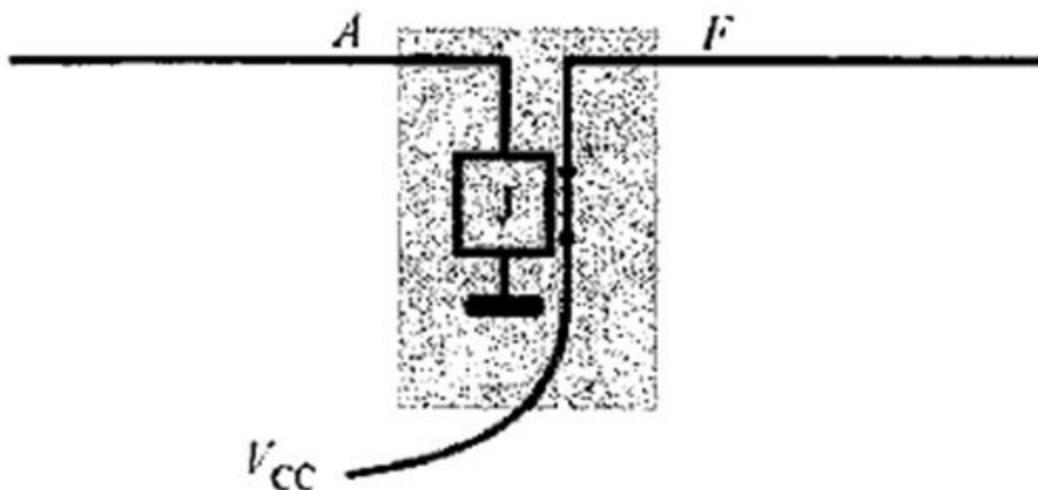


图5.12 非门的构造——一种简单的画法

外部的输入A来自其他设备，比如一个接到电源正极的开关，而且，A必须通过地线流回电源负极才能起作用，而非门的输出F也必须来自于电源的正极。当然这里没有画出F是如何被使用的，这并不重要，重要的是不管是谁使用这个输出F，都应当自行就近接地，这实际上很方便

非门可以用简单的符号来表示（图5.13）



图5.13 非门的符号

非门的符号省略了它需要一个电源这个事实，但这是所熟悉它的人

都知道的。最后，非门实现了逻辑否定，即逻辑非：

$$F = \overline{A}$$

如果特别留意的话，你会看到所有的开关电路都是一些开关串联、并联的组合。所以在讨论了非门后，我们将来到串联电路。像非门一样，一个新型的、电流控制的串联电路通常如图5.14所示，这叫做“与门”，在这幅图的右边是它的符号

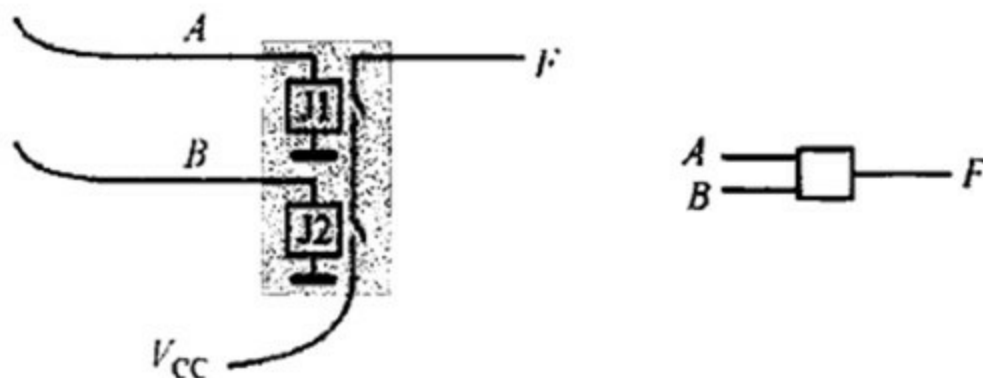


图5.14 与门及其符号（2输入端）

同样是来自命题逻辑的灵感，看得出，它实现的是联言逻辑，即逻辑乘。和普通的串联开关一样，只有当两个输入端A,B同时加电的时候，F端才可能存在输出，在其他任何情况下都不会有输出。

一个两输入端的与门可能会给这里的讲解带来方便，但实际上，一个与门可能有很多输入端，而不仅仅是两个，图5.15就是三输入端的与门

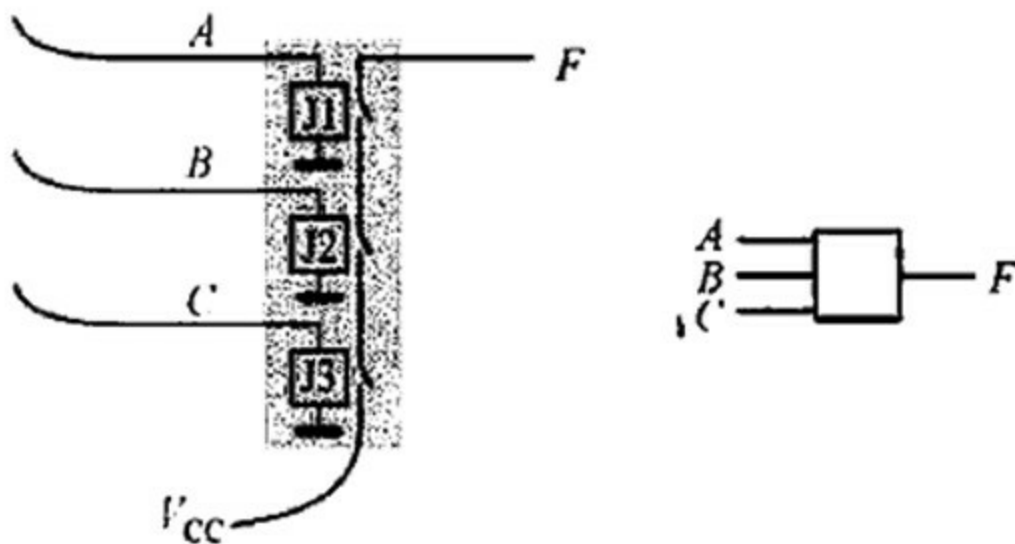


图5.15 三输入端的与门

不过，无论有多少个输入，与门的性质是不会改变的，尽管这里有三个输入A,B,C，但和其他与门一样，除非它们都同时加电，否则F将不



会产生输出  
最后要讲的是并联开关。如图5.16所示

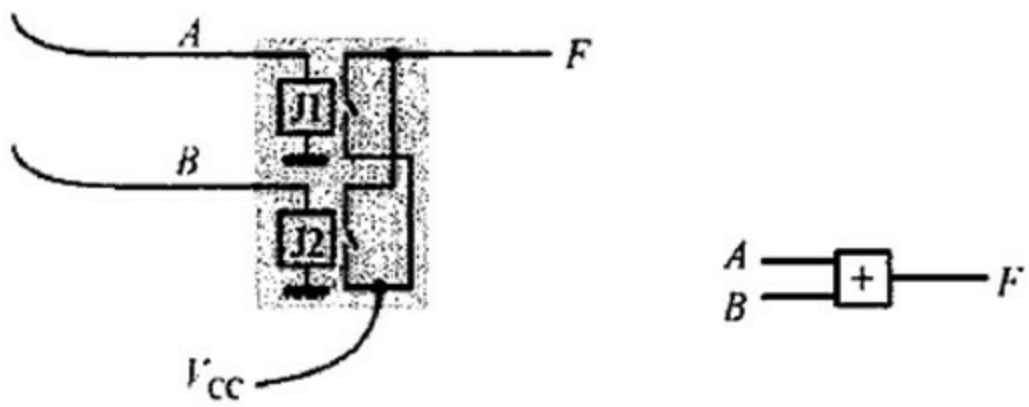


图5.16 或门及其符号

很显然，除非A,B都没有输入，F才没有输出，在其他任何情况下，只要A,B有一个存在输入，或者都有输入，F就一定要有输出，显然这种工作方法符合“或”逻辑法则的

一个这种性质的并联开关称为“或门”（如图5.16右边所示）。和与门一样，一个或门有两个输入端很常见，但并不是一种限制，如果需要，一个或门可以有3个，4个，5个甚至更多的输入端

在了解了与、或、非门这三种基本的逻辑器件后，来看看前面那个大型机电设备的报警电路是如何制作的。同时，这也是一个非常好的例子，可以表明这三种门电路是多么有用。

给定一个实际的开关电路，可以写出它的逻辑表达式，也可以通过直值表来反映出它在不同情况下的状态（比如前面的图5.2）。当然，这只是同一张扑克牌的一面，反过来，对于一个未知的开关电路，即使不知道它的逻辑表达式，如果能够生成一张直值表，也可以得到它。现在，通过这个大型机电设备的例子，我们来实际做一下

因为这台设备有两个输出A和B，而且无论在任务时候，只有当它们都有电，或者都没有电的时候才正常，其他任何情况都意味着麻烦。那么，我们希望开关电路的输出F平时没有电压，只有在A,B不正常的时候才会有电压输出

A,B可能再现的情况只有4种，如果1代表有电压，0代表没有电压，那么A,B和F的关系应当如表5.5所示

表5.5 开关电路的输出和A,B之间的关系

A	B	F
0	0	0
0	1	1
1	0	1

1 1 0

要通过这张表写出开关电路的逻辑表达式，第一步，需要找到输出  $F=1$  的那些行

第二步，对于选出来的那些行，把它们的输入，也就是  $A, B$ ，写成逻辑乘的形式，不过需要注意的是，如果是0，就写成非的形式。

最后，把第二步得到的各项用逻辑加连接起来，整个过程可以用图 5.17 做一个说明

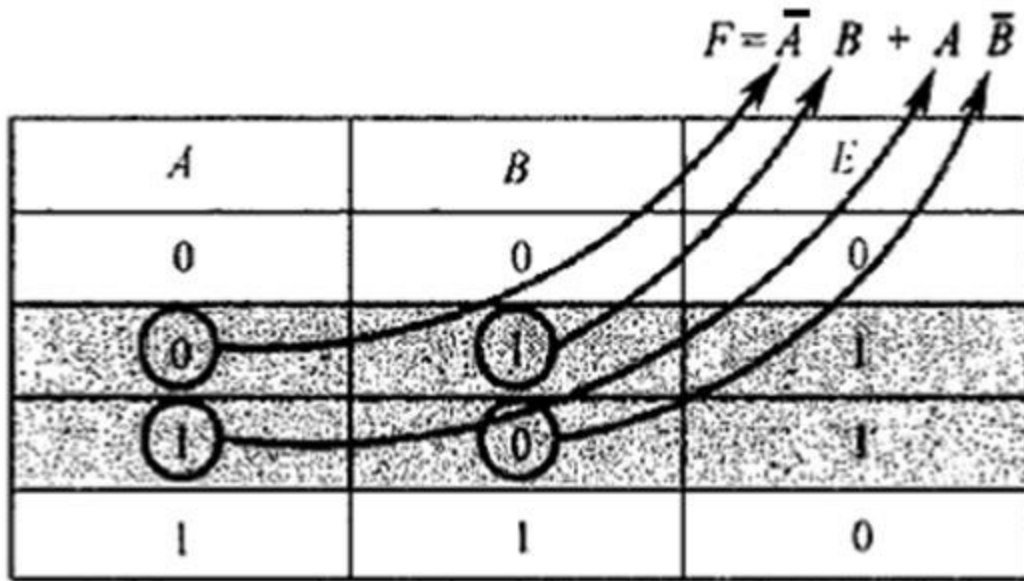


图5.17 用于说明如何从真值得到逻辑表达式的例子

可以得到：

$$F = \bar{A}B + A\bar{B}$$

至此已经学会了如何从真值表得到逻辑表达式的方法，它既简单又有效。如果觉得不可思议，甚至怀疑它是不是真的正确，不妨用所有0与1的组合代入这个式子，看能不能反过来得到上面的真值表。至于为什么要这样作，道理很简单。首先，对于任何一个逻辑与的表达式，只有一种情况会使它为1，比如对于逻辑表达式  $AB$ ，只有  $A=1, B=1$  时， $AB$  才为1.再比如  $\bar{A}B$ ，只有  $A=0, B=1$  时， $\bar{A}B$  才为1，对于  $A, B$  其他任何可能的取值， $\bar{A}B$  都为0

反过来说，要想让  $\bar{A}B=1$ ，只有一种可能，那就是  $A=0, B=1$ ；要想让  $A\bar{B}=1$ ，也只有一种可能，即  $A=1, B=0$

现在的情况是，我们希望一个开关电路在  $A=0, B=1$  或者  $A=1, B=0$  的情况下总是输出1，那么只好采取两头堵的办法，来应付这两种可能出现的情况。好在逻辑加可以解决这个问题：

$$\overline{A}B + A\overline{B}$$

依据逻辑表达式，可以使用前面介绍过的与、或、非门来构造一个实现该逻辑表达式功能的、实际的开关电路。如图5.18所示，先使用两个与门来实现逻辑乘（当然还得先用非门转换一下），然后再把这两个与门的输出送到或门，对它们进行逻辑加，最后输出就是F

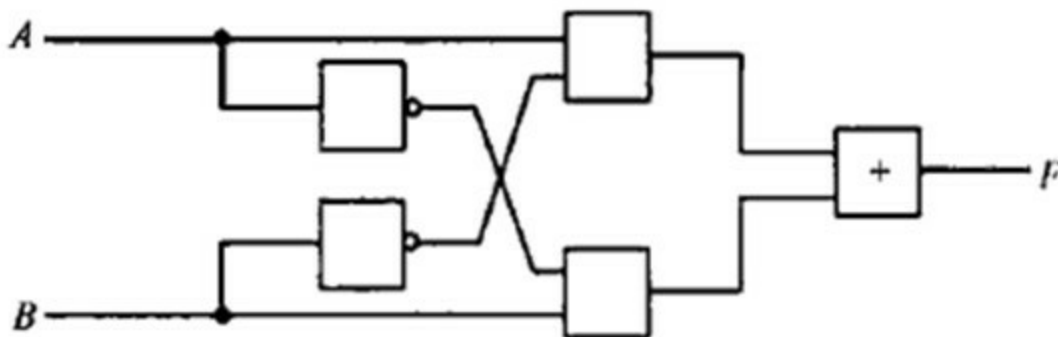


图5.18  $\overline{A}B + A\overline{B}$  的逻辑电路组成

这是一个很有特色的电路，输入端A和B彼此以不同的形式逻辑乘，然后又逻辑加，这称为异或。为了使大家对逻辑门有一个更深刻的认识，现在让我们来看看这个异或电路具体是如何用继电器连接起来的

如前所述，只有当那台机电设备A,B两处都有电，或者都没有电的时候，它是正常的；反之，一个有电一个没电就表明出现故障

说是A,B两“点”以为它就是一个金属探头或一小段裸露的电线，不是这样的，我们知道，电压只存在于电源的正、负两极之间，所以，我们说的A点和B点通常是一根双芯电缆（图5.19）

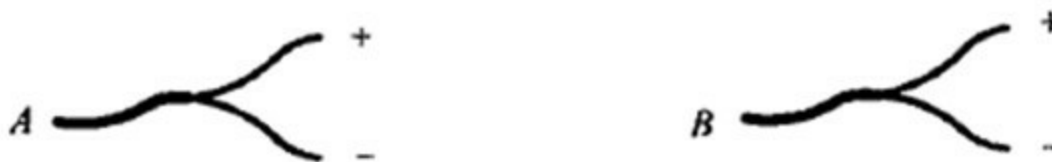


图5.19 所有的电路都应当是闭合的回路，A,B两点要对外供电，就必然是各自包含了两根导线

整个异或电路需要2个非门，2个与门和1个或门，总共需要8个继电器。而且，所有的继电器共用一个电源Vcc（换句话说，这个异或的所有逻辑门都使用同一个电源）。同时，A和B的电压将来驱动异或电路（图5.20）

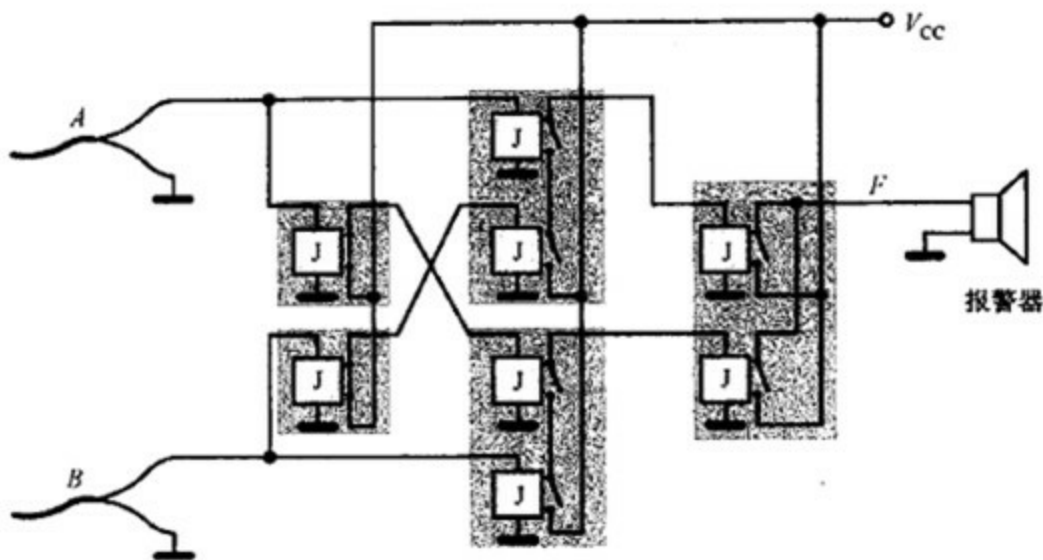


图5.20 用逻辑门来搭建报警电路的完整连接图

这里给出的电路图具体到了每一个与、或、非门内部的继电器，相信已足够清楚。

最后，异或门的输出用来接通报警电路，或者如果后者没有自己的电源，则这个给出用来直接给它供电，使在机电设备不正常的时候开始工作。

异或电路应用得很广泛。在发现了这一点后，工程师们觉得把它做成一个独立的模块可能用起来更顺手，于是一个新的门电路诞生了——异或门产生了。图5.21是异或门的符号（右边是它的逻辑表达式），与其他门电路相比，它显著的特征是中间有一个用圆圈围起来的加号

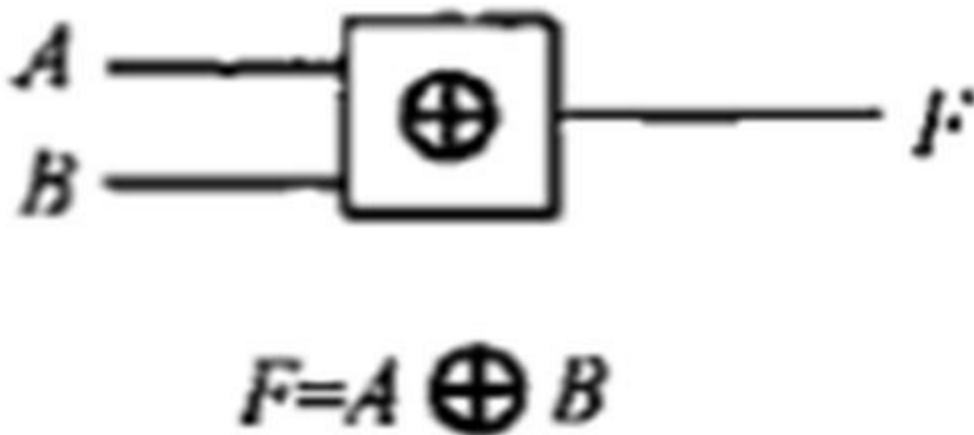


图5.21 异或门的符号与逻辑表达式

从逻辑学到布尔代数，再到香农的开关理论，一直到现在我们能够使用最基本的与、或、非三种门来构造实际的开关电路。

与、或、非是三种最简单的门电路，自从有了它们，什么数码相

机、智能微波炉、智能冰箱、MP3播放器、微型计算机、数字电视，甚至包括我们在本书开头到现在一直在研究的如何制造的加法机，不管现实世界中有多少类似这样的设备，构造它们所用的基本部件差不多很大一部分是这三种门电路。结束本章之前，来认识一下所谓的“莎士比亚电路”，如图5.22所示



图5.22 莎士比亚电路

这个电路没有任何实用的意义，但足以证明计算机工程师们并非都是一些只知道钻研技术而没有幽默感，要是你知道莎士比亚，读过《哈姆雷特》，应该知道这个电路的意思

## 第6章 加法机的诞生

香农的论文《继电器和开关电路的符号化分析》发表于1938年，说到它的意义，只要环顾圆周，就会发现到处是手机、笔记本电脑、便携式音乐视频播放器和掌上电子游戏机，什么数字电视、数字化、数字时代等等，但如果没有香农的开关电路，这一切都不会存在。香农就像第一个学会如何将小麦变成面粉的人，从那以后，世界上开始围绕着面粉有了更多的产品：面包、油条、馒头、花卷、包子、饼干以及其他各种各样面食。同样，香农的理论将指导我们顺利制造出一台加法机

# 6.1 全加器的构造

制造一台加法机的关键是全加器的实现。先回忆一下全加器

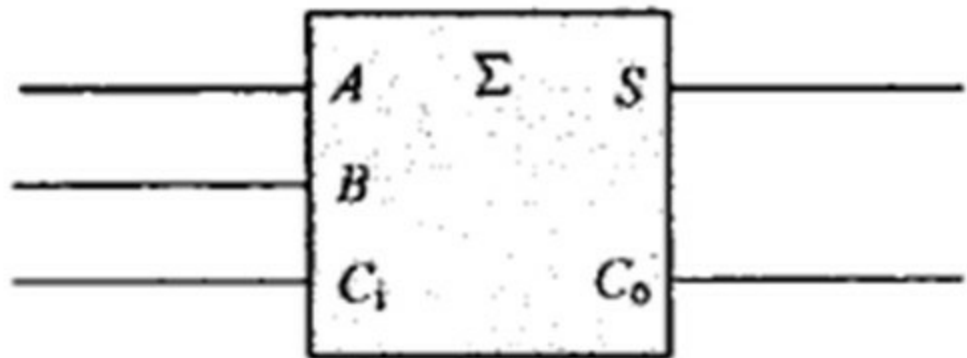


图6.1 全加器的符号

A和B是来自加数和被加数的两个比特， $C_i$ 是来自前一列的进位；S是前面三项加起来的“和”； $C_o$ 是当前这一列向下一列的进位

全加器的复杂之处在于，当它被连到一个电路中的时候，不知道A,B和 $C_i$ 将会是什么，它们是0还是1。所以根据实际情况来安排相应的输出，这就是全加器存在的价值和意义，经过仔细分析，三个比特（A，B和 $C_i$ ）相加有8种可能的情况，分别是：

- 0+0+0
- 0+0+1
- 0+1+0
- 0+1+1
- 1+0+0
- 1+0+1
- 1+1+0
- 1+1+1

全加器操纵的是0和1，除此之外别无他物。这意味着，如果把它和逻辑的真假、开关的闭合/断开相比较的话，会发现如果想要构造全加器，从这些特定的输入得到合适的输出结果，开关电路是一个不错的选择。

对于所有这8种可能的情况，在每一处情况下全加器所产生的“和”与进位分别如表6.1和表6.2所示

表6.1 全加器输出端S的真值表

A	B	$C_i$	S
0	0	0	0
0	0	1	1

0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

表6.2 全加器进位C<sub>0</sub>的真值表

A	B	C <sub>i</sub>	C <sub>0</sub>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

上一章学习了逻辑学和逻辑电路的知识，为了从真值表得到逻辑表达式，实际上我们只需要考虑那些输出为1的行，也就是上面两张表中颜色较深的那些行。这样，从表6.1里把那些使得S为1的行挑出来，写成逻辑表达式：

$$S = \overline{A}\overline{B}C_i + \overline{A}BC_i + A\overline{B}\overline{C_i} + ABC_i$$

接着从表6.2里把那些使得进位C<sub>0</sub>为1的行也挑出来，也写出它的逻辑表达式：

$$C_0 = \overline{A}BC_i + \overline{A}BC_i + A\overline{B}\overline{C_i} + ABC_i$$

到这一步，不需要更多的解释了，已经找到了答案，从根本上解决了制造一个全加器所需的所有技术细节问题，使用的都是最基本的逻辑电路。唯一的不足是所有的与门都是三输入端的，而所有的或门也都是有四个输入（图6.2）



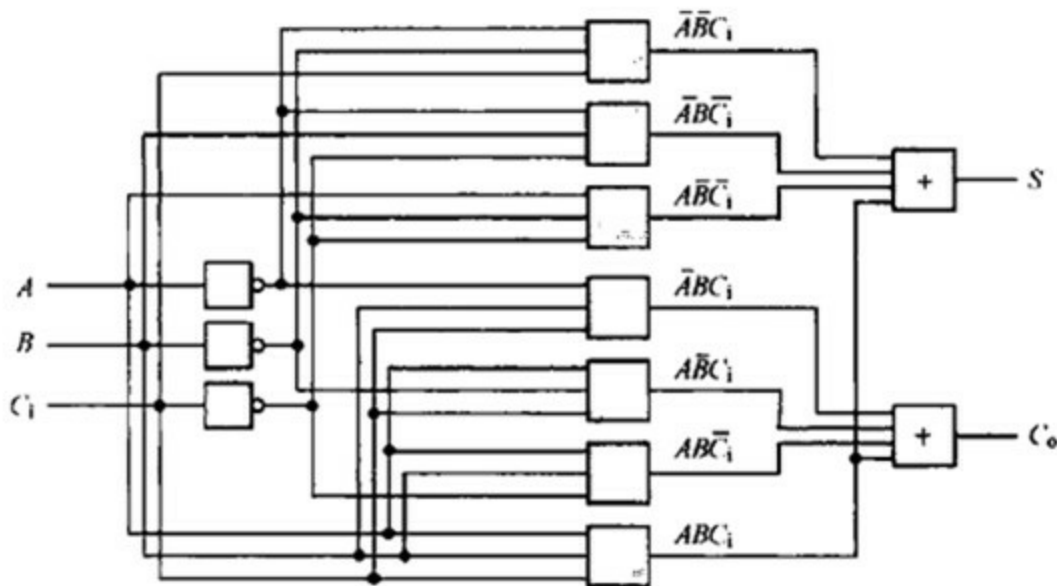


图6.2 全加器的逻辑电路实现

从图中可以看出，这需要大量的继电器，由于每个非门需要1个继电器，每个三输入的与门需要3个继电器，而每个四输入的或门需要4个继电器，所以制造一个完整的全加器需要32个继电器。这还只是一个全加器，要知道，一个完整的、能计算大数的加法机需要一大堆全加器。

好在有很多逻辑表达式可以通过简化得到更简单的形式，就像做代数题一样。在用逻辑门构造电路的时候，通过简化逻辑的表达式，可以节省很多材料

如何简化逻辑表达式，这是个很有趣的话题，需要几个规则、若干条定理。要想掌握它，得在大学里听课，或者找一本数字逻辑和逻辑电路的书好好读一读。这里有个小例子，比如：

$$A+AB$$

它可以像普通代数运算那样表示成：

$$A(1+B)$$

但逻辑表达式与普通代数运算的相同之处到此为止了。由于逻辑表达式的工作是计算逻辑上的真与假，所以不管B是0还是1，下式都成立：

$$1+B=1$$

所以得出结论：

$$A+AB=A$$

换句话说， $A+AB$ 的值其实与B无关，这就是逻辑表达式化简的一个典型例子。相似地，前面那两个全加器的逻辑表达式：

$$s = \overline{A}\overline{B}C_1 + \overline{A}B\overline{C}_1 + A\overline{B}\overline{C}_1 + ABC_1$$

$$C_0 = \bar{A}BC_i + A\bar{B}C_i + AB\bar{C}_i + ABC_i$$

可以化简为：

$$S = A \oplus B \oplus C_i$$

$$C_0 = C_i(A \oplus B) + AB$$

这样就可以使用异或门来重新制造全加器，如图6.3所示

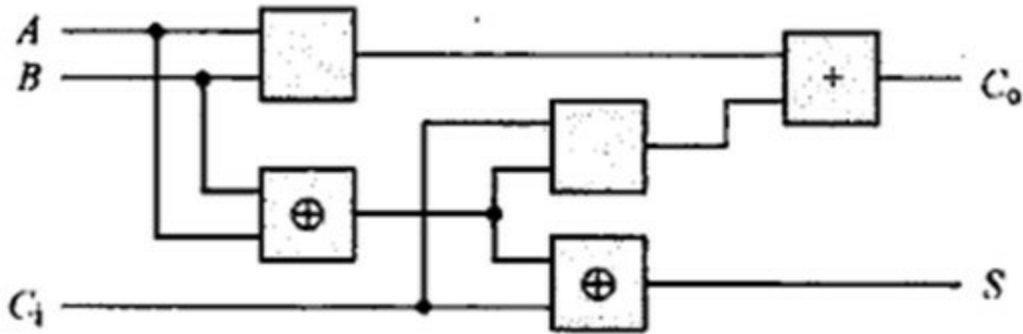


图6.3 用异或门组成的全加器

2个异或门、2个与门、1个或门，使得继电器的总数减至22个，当然，如果继续简化逻辑表达式，可以将每个异或门所使用的继电器数从8个减少到6个，这样一个全加器实际上只需要18个继电器。上面那两个逻辑表达式是如何化简的，以及如何将异或门的继电器数从8个减少到6个，需要你自己抽时间在本书之外慢慢探索了

最后表6.1和表6.2左边第3列在每一行都是一模一样的，列出的都是A,B和C<sub>i</sub>所有可能的组合。通常，这两张表可以合起来，只用一张表反而显得更清楚（表6.3），但要注意S和C<sub>o</sub>要分开处理

A	B	C <sub>i</sub>	S	C <sub>o</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

一个全加器只能计算两个一位二进制数的加法，用处有限，要计算更大的数，比如101+110,则需要用多个这样的全加器互相连接，以形成一个完整的加法机，如图6.4所示

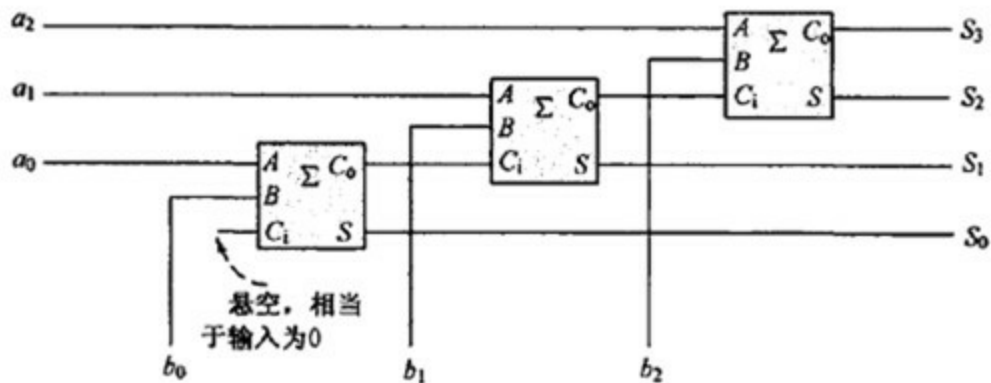


图6.4 用全加器组成一个三比特加法电路

我们曾经在前面见过这个图。两个二进制数 $a_2a_1a_0$ 和 $b_2b_1b_0$ 分别是被加数和加数，而 $S_3S_2S_1S_0$ 则是加法的结果。当然，它只能计算3位二进制数的加法，如果想要计算更大的、更多比特的二进制数，就要使用更多的全加器

## 6.2 加法机的组成

当把所有的全加器连接在一起、封装起来的时候，我们就看到一个完整的加法机（图6.5）



图6.5 加法机的简单图示

要想使这个加法机真正工作，需要用一些开关从电源取电，为它输入两个二进制数。同时，还可以将所有的输出和灯泡连接起来，这样就能直观的看到计算的结果（图6.6）

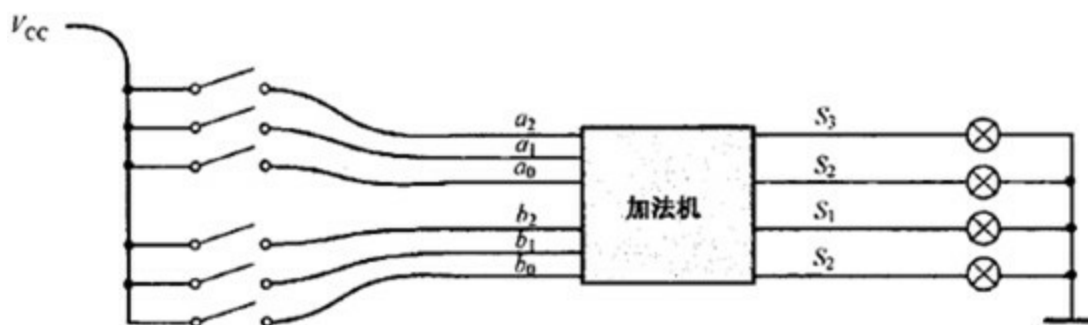


图6.6 加法机电路的完整连接图

很明显，整个电路使用的是一个电源，电流从正极通过各个开关流入加法机，开关的通断决定了各个比特是1还是0，另外还要注意，加法机内部的各个全加器（要是分得更细一点的话，应是全加器内部的每个逻辑门；或者再细一点，是每个逻辑门内部的继电器）也都需要电源供电，但我们省略了这些细节

除了要数量可观的继电器外，这样的加法机在工作时很壮观，随着开关的断开与闭合，灯光闪闪。除此之外，你也会惊奇的发现，只要接通电源，这台加法机随时都在工作 – 就在你摆弄那些开关、将它们置成最终那个数字的过程中，它也在计算，因为每断开或闭合一个开关，就会生成一个新的数字

这本书读到这里你已经可以动手制作一台属于自己的加法机了。1937年11月，美国贝尔实验室研究人员乔治·斯特彼兹（George Stibitz）制造了电磁式数字计算机Model-K，几乎在相同时期，德国工程师楚泽

也独立研制出二进制数字计算机，有趣的是，斯特彼兹的Model-K和楚泽的Z-3计算机采用的元件相同，都是使用电话继电器

Model-K是世界上第一台能做四则运算的计算机

斯特彼兹（1904-1995）出生于美国宾夕法尼亚州约克市，从事的专业方向是数学和物理，从1937年开始在著名的贝尔实验室从事研究工作。贝尔实验室以电话发明人贝尔的名字命名，当时主要研究改进电话的通信性能，斯特彼兹的工作恰好和电磁有关，而且不可避免地要经常与继电器打交道

我们现在几乎可以肯定，是继电器激发了斯特彼兹的灵感。同年，他想到了用继电器来制造一台可以计算数学题的机器，继电器的吸合与断开恰好对应着0和1这两种状态。斯特彼兹一心想把这台机器造出来，他甚至把零件带回家，在厨房的餐桌上组装。他在实验室取得几只继电器，从空铁皮罐头盒剪下两片铁皮作为“输入设备”，又找到几只手电筒灯泡充当“输出设备”，当他把所有元件固定在一块三合板上，这台计算机的装配过程就大功告成。无论从哪个角度看，斯特彼兹的“伟大发明”都像某个中学生完成的一项科技小制作。



就是这样一件模型，斯特彼兹却用它完成了两位二进制加法运算。他已经跨过了一个时代 – 不仅实现了从机械式计算机向电磁式计算机的飞跃，而且制造了一台真正的数字计算机！

他的夫人不无揶揄地建议叫做“餐桌”（kitchen table）。斯特彼兹接受了这个建议，将其命名为Model-K（K就是Kitchen table）

第二天，斯特彼兹捧着他的“宝贝”向实验室里的同事介绍，实验室里的同事不以为然，实验室里有手摇计算机，不需要这种机器。

斯特彼兹又花了几个星期来改进Model-K，机器的性能越来越完善，只是很长时间内，仍然没有人理睬他的机器，直到有一天，数学研究室主任问他“你的Model-K计算机能不能帮我们解决复数计算的难题？”

贝尔实验室面对的问题是交流电路实验中需要给出答案的大量复数计算题，实验室雇佣了许多女计算员，用手摇计算机从早到晚的计算，仍然跟不上实验进度

斯特彼兹给了肯定的回答，正式研究数字计算机的项目因此获得了新的转机。贝尔实验室为他配备了助手，包括一位电气设计师威廉姆斯（S. Williams）。1938年9月，命名为M-1的数字计算机研制工程启动；一年之后，即1939年9月，斯特彼兹交出了满意的机器。1940年1月8日，M-1开始运行，标志着美国第一台数字计算机诞生。

M-1电磁式计算机只使用了440个继电器和10个闸刀开关，就完全解决了复数的加、减、乘、除四则运算，一次复数乘法需要30-45秒，计算同样的题目人工手摇计算机需要15分钟。

斯特彼兹在攻读博士学位前，曾在一家电气公司打工，被派到郊区农场去进行无线电测试。每天早晨赶去上班，农场木屋里很冷。他和同事就制作了一台小型遥控器，能自动控制壁炉风门上的开关，这样一来，当他们清晨赶到农场上班前，就能在路上遥控打开风门上的开关使壁炉升温。既然壁炉风门能够遥控，他想试一试遥控M-1计算机

他首先在曼哈顿的办公室里，在不同的房间分别安装三台电传打字机，用电话线与M-1相连，试验很成功。9个月后，电话线已经连上了远在新罕布什尔州的第四台电传机，距离达到250英里

1940年9月，美国数学会在达特默斯大学召开学术会议，会议期间，斯特彼兹派人前往，向包括冯·诺依曼等演示如何遥控M-1计算机，这次成功的演示，在计算机发展史上具有特别重要的地位，它标志着人类已经实现了远程控制计算机。

从1940年到1949年，斯特彼兹接着进行了M-2,M-3,M-4,M-5型电磁式计算机的研制，以满足美国在二次世界大战和战后恢复建设对计算机的需求。M-5占地200平方米，有近万只继电器，共生产了两台。1949

年，贝尔实验室的最后一台M型计算机M-6投入使用，用继电器来组装计算机从此成为了历史

斯特彼兹被誉为“数字计算机之父”

“集成电路之父”基尔

“微处理器之父”霍夫

“鼠标器之父”道格拉斯

“因特网之父”塞尔夫、卡恩

“万维网之父”伯纳斯-李

“电子邮件之父”汤姆林森

## 第7章 会变魔术的触发器

从某种意义上说，计算机就是开关。我们用继电器开关做成了加法机，一台加法机的构造说白了其实就是一大堆开关的精巧组合，而人们之所以能够进行这种组合，利益于物理学、数学和逻辑学方面的最新进展，以及某些先行者将它们融合的非凡才能。

通常开关的作用是非常直接的 – 非通即断。接通开关，灯泡亮了、电动机转了、电视开了……；断开开关，灯泡灭了、电动机停了、电视关了……尽管我们刚刚用这种类型的开关造出了一台加法机，但要制造一台真正现代的、功能强大的计算机，仅仅有这些简单的开关是远远不够的。所以，在这一章，我们将学习如何来制造一些特殊的开关，这些开关不再是我们在生活中经常接触并司空见惯的那些开关，这些东西对制造现代的计算机至关重要。



## 7.1 不寻常的开关和灯

一般来说，开关的作用是很直接的，接通开关灯就亮了；断开开关灯就灭了。开关和灯泡之间的关系似乎一起就是这样。

下面是一个不可思议的电路，它左边连着两个开关A,B，右边有一只灯泡，如图7.1所示



图7.1 连着两个按键开关的逻辑电路

为了更直观地说明这个电路所发生的一切，这里使用了按键开关，这种开关不同于我们平常看到的那些开关，当你按住它的时候，它是通的；一放开手，它又断开了，和计算机键盘或手机上的按键一样。而我们以前使用的开关则是合上时是通的，想让它断开，还要再动手把它扳开

假设一开始灯泡是灭的，那么按下开关A，灯泡亮了。这很自然，可以理所当然地把它解释成开关A控制着灯泡，接通A，灯泡有电流通过，灯泡亮了。想当然地认为如果断开A，灯泡就会灭——但当手松开时，开关A断开，灯泡依然亮着！再按下A，灯泡依然亮着

这已经很奇怪了，还有更奇怪的。通常情况下，接通开关的时候灯才会亮，因为这时才有电。可是当按下开关B时，灯泡居然灭了！再按B，灯泡依然不亮。

情况就是这样，无论什么时候，按一下A，灯亮了，再按A，灯还是亮的；按一下就灭了，再按B灯泡依然不亮

真奇怪！这是怎么回事呢？这个电路是怎么做到这一点的呢？

## 7.2 反馈和振荡器

继电器用来制造加法机，要想知道它还有什么用，这里有一个比较好的实例，取一个继电器、一只灯泡、一个开关，用电线把它们按图7.2那样连接起来

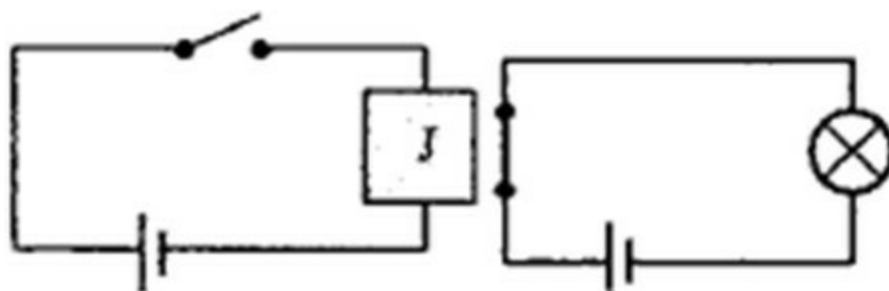


图7.2 用开关和继电器控制另一个电路的通断

这里使用的是一个常闭触点式的继电器，内部的衔铁开关处于接通状态。当左边的开关断开时，右边的灯泡是亮的；反之当左边的开关闭合时，继电器的磁力把衔铁拉开，灯泡就灭了

在本书的第5章，画一个复杂的电路图时，用不着把电池画出来，也用不着把每一根连接到电源负极的线都画出来，用一种更专业的方法来画上面的电路，如图7.3所示

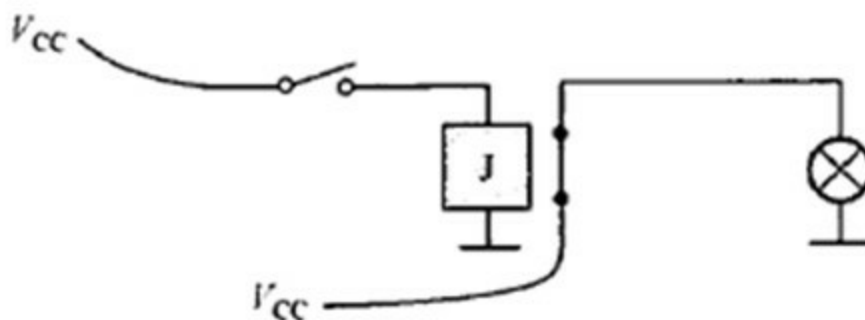


图7.3 继电器电路的另一种简化形式

很明显，该电路的工作就是将输出变得与输入相反。这东西似曾相识，是的，常闭触点的继电器就是一个非门，所以，图7.3可以进一步简化成图7.4

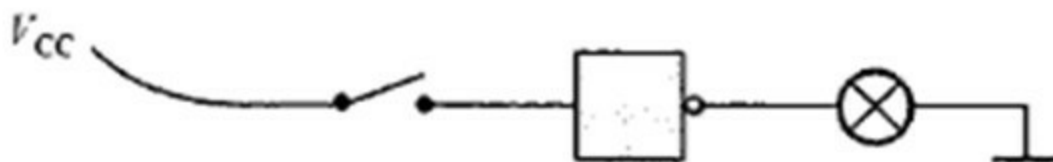


图7.4 带电源的常闭触点继电器（即一个非门）

我们把图7.3中输入连同那个开关统统去掉，直接用它的输出作为

输入，图7.5所示

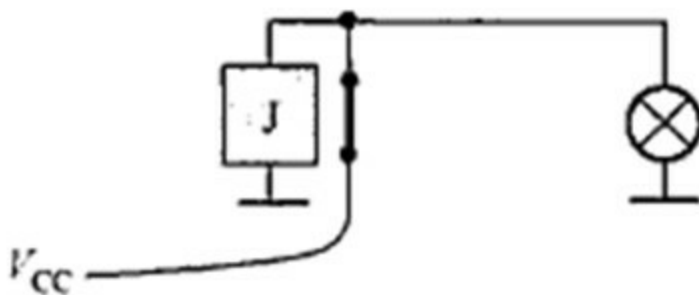


图7.5 继电器的输入与输出共用一个电源时的不同情况

这回，因为继电器的衔铁平时是闭合的，当它一通电，会立即点亮灯泡，不过，这个输出又是继电器的输入，所以在灯泡亮的同时，电磁铁产生磁力，把衔铁开关拉开，于是灯泡灭掉。同时电磁铁也失去磁力，于是衔铁开关又恢复原状，将电路接通，灯又亮了。

就这样，只要电源有电，这个经过特殊连接的电路将一直工作在瞬间有输出、瞬间无输出的状态，而灯泡一亮一灭

但本质是依然是一个非门，只是连线有些特殊，是一个首尾相连的非门，如图7.6所示：

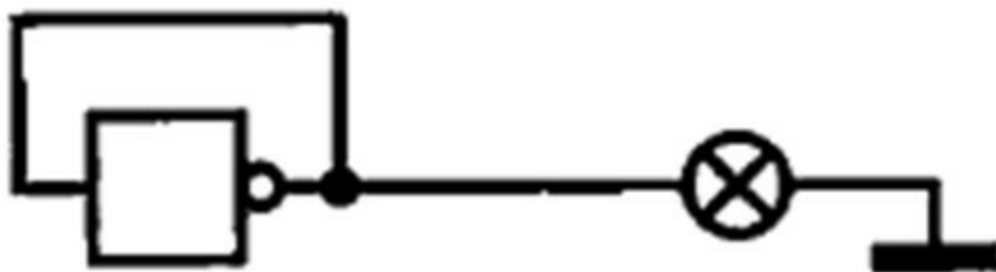


图7.6 把非门的输出和输入相连构成一个振荡器

把一个非门的输出取出一部分来，同时又作为它的输入，这样就形成了一个反馈。

以前我上学的时候，学校的闹铃就是使用一个首尾相连的非门电路，只不过把继电器的衔铁换成了一个小锤子，旁边是一个大铃铛。

一个非门，再加上反馈后，就能产生一连串交替变化的输出，使得与之相连的灯泡一亮一灭，很像一把振动的直尺或一个来回晃动的秋千，在两个端点之间来回运动，作为一个类比，像这种东西，在电子技术领域叫做振荡器。

发明者往往有将他的成果用到极致的愿望，如果笛卡尔还活着，他一定想看看振荡器的输出在他的坐标系中是什么样子，现在只有我们替他做这件事了

一开始，非门是有输出的，我们用一条直线来表示，并且把它画在纸上靠上的位置，我们省略了坐标轴，因为输出的电压具体有多少伏并

不重要，持续的时间（线条的长短）也不重要，不过很快，由于反馈的关系，非门失去了输出，这意味着输出为零，在刚才那条横线的下面，也就我们认为是零的地方再画一条横线，表示当前没有输出，或者说输出是零伏，并且也用线条的长短代替持续时间，再往后，这个振荡器一直工作，而它的输出也必然如图7.7那样交替变化下去

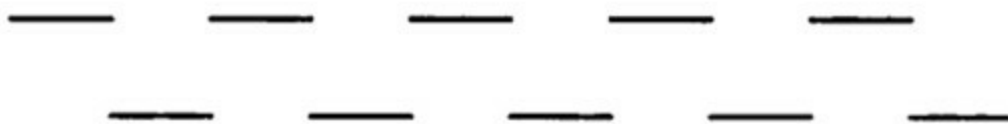


图7.7 非门振荡器的输出是高低交替的

实事求是地说，这不是一个真实的振荡器的输出图像。首先，由于继电器内部开关是机械的，而所有的机械开关都有一个毛病：在接通和断开瞬间会发生抖动或者说震颤，所以它的波形一开始像锯齿，但极其短暂，然后才稳定成一条直线。所有的机械开关都存在这样的总是，但我们这里可以无视它的存在，它对我们当前讨论的主题没有影响。

其次，像每次打开或关闭水龙头一样，电路在接通或断开的瞬间，电压或电流不会马上就达到最大值/最小值，总有一个从小到大或从大到小的过程。如果是那种老式旋转开关的水龙头，这种现象就特别明显。对于电路来说，造成这种现象的原因是多方面的，而且这个变化过程通常也极其短暂，由于现代科技的进步，这个变化过程可以缩短到几纳秒（1秒=10亿纳秒）或者更短暂。为了表示这个变化过程，需要在两条输出线条之间添加“坡度”，即一条非常陡峭的斜线。但由于这个过渡太快太陡了，看起来人们更喜欢为图方便而直接把它画成一条竖线，如图7.8左侧所示



图7.8 振荡器脉冲的上升沿和下降沿

这是一种非常有规律的、周期变化的波形，属于有棱有角的方形，因此在电学里叫做“方波”。类似于我们人体那样有规律跳动的脉搏，也许因为这个原因，现在讲的这个振荡器，它所产生的方波被称为“脉冲”。对于人类来说，脉搏的跳动表明我们还活着，而对于计算机来说，脉冲的意义也同样如此，如果没有脉冲，计算机就完了

对于振荡器输出的每一个方形脉冲，电压或电流从零上升到最大值的那条线叫做上升沿；反之，电压或电流从最大下降到零的那条线叫下降沿，如图7.8右侧所示。很清楚的是，在图7.8中，左边的图形实际上是右边图形的简单重复，和其他有规律变化的波形一样，每秒能产生多少个这样的脉冲，称为这种振荡器的频率。回忆一下第4章中频率的说

明，频率是1秒之内相同波形出现的次数，在这里就是指每秒能产生多少个图7.8右侧那样的脉冲，平均下来，每个脉冲所占的时间，或者更准确地说，两个脉冲相继出现的间隔时间就是脉冲周期，它是频率的倒数，就当前的例子来说，如果每秒出现5个脉冲，那么频率就是5Hz，周期为0.2秒

精确的振荡器（即每秒产生的脉冲个数非常准确和稳定）应用十分广泛，直到现在，还有很多人在用一种需要安装电池才能走动的钟表，当然它还能定时，在这种钟表里面，有一个振荡器（用的当然不会是这种简陋的继电器），它每隔一秒产生一个脉冲，用来驱动一个小电动机，促使电动机转动一个角度，并通过齿轮传动机构带动秒针跟着向前移动一步，并“嗒”地响一下，由于这个原因，这种振荡脉冲经常被称为时钟脉冲，或者时钟信号

## 7.3 电子管时代

振荡器的发明最早的目的是为了向天空中发射电磁波，在20世纪之前人们就知道，要想产生电磁波，必须使用电流以极高的频率不断变化，而要产生高速变化的电流，振荡器可以做到这一点。

世界上第一个振荡器出自赫兹之手，也就是前面讲过的那个赫兹用于证明电磁波存在的装置，它每冒一次电火花就会向外辐射电磁波。

20世纪初是一个激动人心的时代，那时已经有了电灯、电话、电报、留声机，无线电技术也在起步。毫不夸张地说，整个20世纪的前十年就是现代信息技术的黎明阶段或者说是第一缕曙光出现的时候。新的发明不断出现，而这些发明又造就了更多的发明

电磁波的发现使科学家非常激动，因为他们想借助它来传递声音，至于动机，应该有两个：第一，不需要架设电线，节省材料和成本，尤其是对于不能或不容易架线电线的地方来说，这种好处尤其明显；第二，这是科学工作者的本能。他们根本不会知道自己的工作会让一百多年之后的我们用上了手机，相反的，他们只是想电磁波能不能传递声音。

愿望很美好，只是在当时不可能实现。原因很简单，而且有两条，第一，没有一种好的方法把来自话筒的声音电流加载到电磁波上，而且在接收方也没有办法将微弱的信号放大；第二，当时的电磁波发生装置都很原始，电流通过空气放电时，是一个没有规律的导电过程。这意味着，不规则的急速变化的电流，将产生包含各种频率成分的电磁波，这种电磁波是不“纯净”的。我们需要的是那种波形非常规则，并且只在单一频率上工作的发射装置

那么我们刚刚发明的振荡器不就很好吗？而且当时已经有了继电器  
答案是不好，原因很简单，发射电磁波需要很高的振荡频率，而机械继电器显然不能胜任。我们平时用收音机听的广播，频率在500-1600kHz之间；电视节目需要几百兆赫兹的频率，而手机则更高为几GHz，继电器是机械装置，每秒断合几下，几十下可能还行，要让它每秒断全几百万下，这是不可能的。

1kHz=1千Hz, 1MHz=1百万Hz, 1GHz=10亿Hz

所以，如果要发明一样东西，可能需要事先发明另外一些东西

1904年11月16日是电学史上一个很重要的日子，在那天，一个名叫弗莱明（John Ambrose Fleming）的英国人发明了一种新鲜玩意儿，这东西说起来真是很简单，其实就是一个灯泡，也就是说，它是一个被抽成真空的玻璃瓶，里面装上灯丝，通电可以灼热发光。

当然，它肯定不会仅仅是一只普通的灯泡，要不然也不会称为它为

一项发明了，因为美国的爱迪生早已发明了灯泡。在这个玻璃瓶中，有一根导线安装在离灯丝不远的地方（但没有接触），隔着一定距离，再安装上另一根导线，如图7.9所示

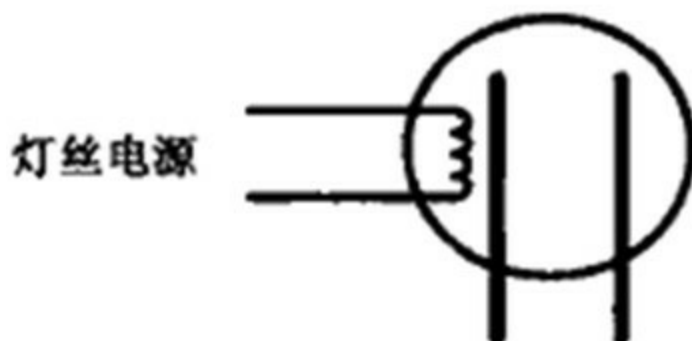


图7.9 整个20世纪的电学成就始于这个简单的发明

这个装置没有什么特别之处，除了里面多了两根电线，显得有些古怪之外，它和一只真正的灯泡没有什么区别

这个世界很奇妙，平平常常的事物往往隐藏着玄机，取决于你是否有心。现在，我们再拿一个灯泡 – 真正的灯泡，然后，再用一个电源按图7.10所示那样连接起来



图7.10 这是一个具有单向导电性的发明

通常情况下，右边那只灯泡是不会亮的，这符合我们的常识，因为没有电流通过，因为右边的那个奇怪的灯泡即使被一个抽成真空的罩子罩着，但两根电线是分开的，相当于断路

此时，给左边的奇怪的灯泡通电，让它灼热发光，这回，在真空中，两个隔着一段距离的导线间竟然有电流通过，右边的灯泡居然亮了。这真的太奇怪了，以至于有个富有想象力的作家把这种现象称为“真空驯电子”，很贴切也很形象

这已经足以让人目瞪口呆了，但更令人惊讶的是 – 这也是它特别有用的原因 – 如果把电源的正负极对调一下，让离灯丝近的的那根导线接正极，离灯丝远的那根导线接负极，这回电流却消失了

这事儿肯定和灼热的灯丝有关，因为只有它在通电灼热时才会发生这样的情况，但不一定是非常直接的关系。毕竟，在灯丝和接电源负极（离灯丝较近的那根导线）之间还有一点儿距离，也就是说，电子不是

从灯丝那里来的

这在当时，具体的原因弗莱明也不清楚。现在我们知道，要是在两根金属之间加上电压，被灯丝灼热的金属可以向真空中发射电子，有点儿像太阳风，那个极热的巨大球体源源不断地向整个太阳系抛射高能粒子，或者换一种比喻，因为导线离灯丝很近，会被灯丝加热到很高的温度，而高温的导线就像工厂里用的那种风力强劲的电风扇，甚至可以吹动一个人，但要逆风靠近它却很困难。电子其实是从负极（阴极）出来，流回正极（阳极），而不是长期以来一直认为的那样从电源的正极出来流到负极

弗莱明用来做实验的装置和我们现在描述的不太一样，其中最大的区别就在于他的玻璃瓶内没有那根靠近灯丝的导线，而是直接用灯丝来代替那根靠近它的金属，但原理和实际效果一样

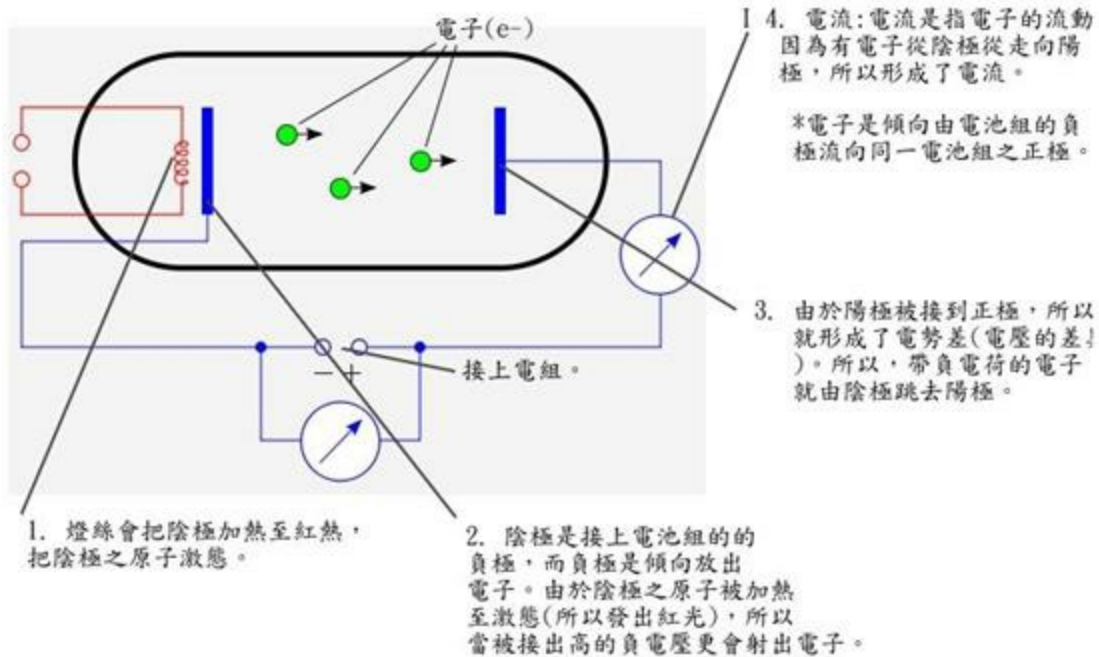
这个装置对电源的接法很挑剔，用术语来说就是具有单向导电性。它还有一个专业名称 – 电子二极管，毕竟，它真正有用的是那两根被分隔开的电极，而灯丝则不算在内。为了便于说明，我们把靠近灯丝的那根金属称为阴极，因为它通常要接在电源的负极上（负极也称阴极，不管在工程还是生活中，这两个名称交替使用，说的都是一个意思），主要的作用是发射电子；另一根金属叫做阳极，通常接在电源的正极上，用来把电子从阴极吸引过去。还有两个引脚是用来给灯丝供电的。如图7.11所示，是电子二极管的符号



图7.11 电子二极管的符号

注意观察，阴极被画成一个半圆，这不是为了显得美观而特意画的，而是当人们了解了电子二极管的工作原理后，为了使阴极能更有效地被灯丝加热，把它做成一个筒形，像小桶一样把灯丝罩在里面，而且，在这个小桶的底部涂了一层氧化物，比如氧化铜，这是因为氧化物在加热后，发射电子的本领更高。

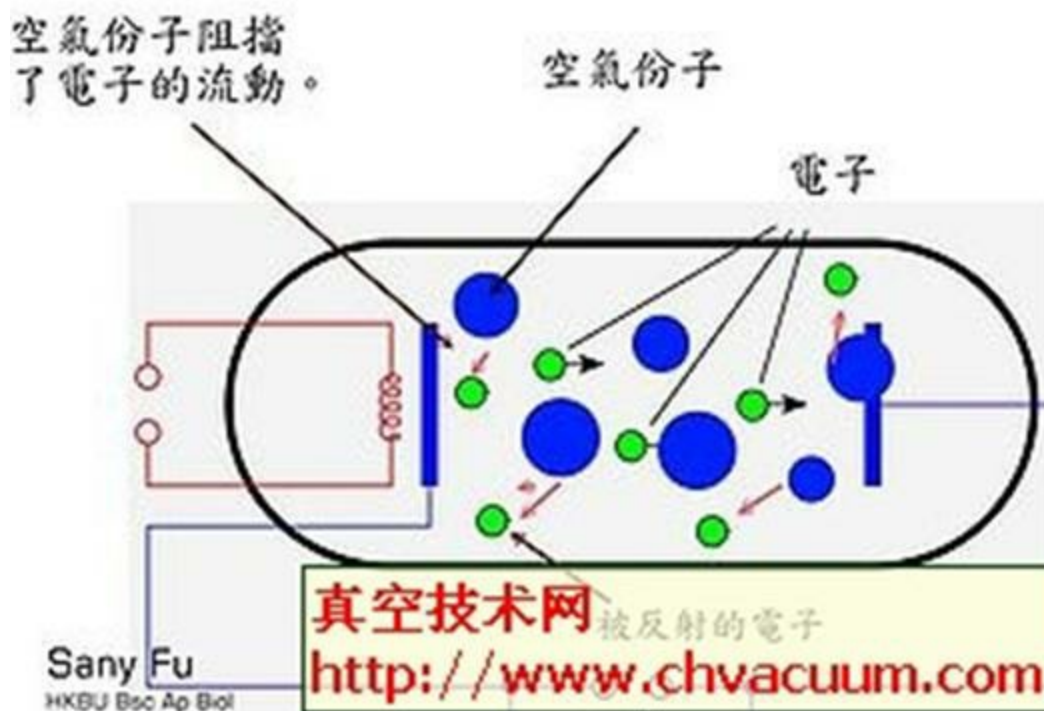




## 真空二極管之簡化原理

## 真空管内需要被抽成真空的原因

电子在其放射过程中，因会与组成空气的分子碰撞而产生阻力，因此电子经由空气之类的介质来移动的话，将会比在真空状态困难，所以要想轻松地实现电子放射的移动过程，需要将产生电子放射及电子收集的各项元件，也就是灯丝、阴极、栅极、屏极等封装于玻璃管内，且将其内部抽成真空，才能使电子的放射运作效率最高，而此玻璃管也就是所谓的真空管。若真空程度不高，更会发出蓝光和严重影响真空管的工作表现。发出蓝光的原因是被阴极射出的电子碰撞电子管中的空气分析，使空气的原子被激发到激发态。



弗莱明发明了电子二极管按通常的说法最初的灵感来源于爱迪生，爱迪生在发明灯泡的过程中曾经发现过这种现象，作为一名科学家，爱迪生本能地意识到这很不寻常，但他忙于制造灯丝，根本没时间深究这个现象，他只是把当时的情况记录下来。

那个时代是电学大发展的时期，人们乐于尝试，这样，在大致明白了电子二极管的原理后，他们开始向里面加入一些别的东西 – 通常是金属 – 看看是不是会发生一些别的有意思的事情。

在这方面干得最起劲的人是美国人福雷斯特（Lee de Forest）。据说他小时候就喜欢摆弄各种机械。

电子二极管的发明使福雷斯特很失落，因为他一开始和弗莱明一

样，受了爱迪生记录的那个现象的启发，想进行发明，不过弗莱明抢先发明了电子二极管。失望的福雷斯特开始摆弄电子二极管，希望能为它做些完善的工作，很自然地，人们会想，既然电子是从阴极通过真空流向阳极，那么，能不能控制它的流量，同时，也许能够看到一些古怪的事情发生。总之不试试谁知道呢

为了控制电子从阴极到阳极的流动，福雷斯特在原有的电子二极管里，也就是阴极和阳极之间，又加了一根金属丝，后来改成金属网，之所以做成网状，是因为既能够让电子容易通过，又可以对它们施加控制，很像我们平时看到的栅栏，所以称之为控制栅极，也叫栅极，如图7.12所示



图7.12 电子三极管的符号

历史证明，福雷斯特笑到了最后，他的锲而不舍最终给自己带来了好运气，也使他名留史册。在这个装置上，他给阴极和阳极供电，就像电子二极管那样，同时，也给阴极和栅极供电，如图7.13所示

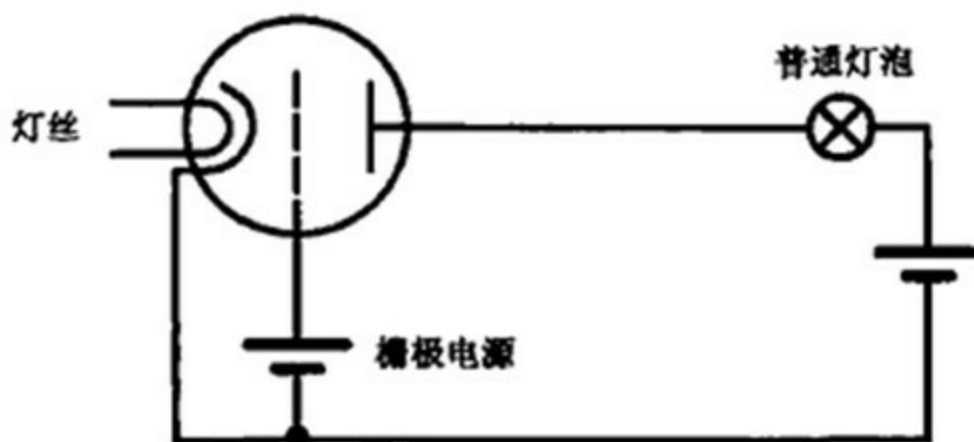


图7.13 电子三极管原理示意

通过改变栅极电压的大小和极性，可以改变阳极上电流的强弱，甚

至切断电流，这叫截止。

这的确很有意思，但差不多是意料之中，似乎没有什么新意。不过，令人意想不到的，只要栅极上的电流发生一点点变化，阳极上的电流就会大幅度地跟着改变。比如上图，细微地调整栅极电源，就会明显地改变灯泡的亮度。

这意味着因为比电子二极管多长了一条腿，电子三极管具有放大作用，所以这在电学史上是个很重要的事件。要知道如果没有这个开端，恐怕我们现在还听不上音响、看不成电视.....

注意电子三极管的放大效果不是无端凭空产生的，这种放大后的能量来自于电源，它只是一个转换器

这个发现的意义非同寻常，首先对于微弱的电话信号可以使用电子三极管放大，要是一只电子三极管放大倍数不够，还可以多加几只进行接力放大。

除了有线的东西需要电子管，对于无线来说，利用三极管的放大作用，再加上适当的反馈，就可以形成一个振荡器，能够产生固定频率的振荡电流，如果振荡频率足够高，就能向很远的地方发射无线电波，而且它的优势是可以获得极高的振荡频率，因为电子管的开关速度很快。

这是真正的振荡，不但辐射电磁波的频率高了，而且只在固定的频率上工作，除非接收器希望接收这个频率上的信号，它不会对其他频率的接收器产生干扰，尽管天空中的无线电波越来越多，不会互相干扰。事实上，也就是从这个时候开始，利用电磁波进行语音和电报通信的时代开始了。

从电子管发明到现在，全世界生产的电子三极管太多了，人们称赞福雷斯特“推动了无线电技术的迅猛发展，引发了一场革命并奠定了近代电子工业的基础”，福雷斯特被尊称为“无线电之父”。

# 真空管

## 电流与电子流动

真空管当然不是无缘无故地做几片金属板封装在抽成真空的玻璃瓶里进行试验的，它的发明与爱迪生有关系。

电流与电子流动的方向恰巧相反

过去的科学家无法观察电子移动的方向，于是规定，将电池的某一极设定为正极，其电压为正电压，电流由正极流向负极而形成一个封闭的回路。因此多年来没有发生任何冲突，直到近代科学家有了更精良的设备，观察之后推翻了之前的说法：原来电子是由电池的负极流出来的爱迪生效应

爱迪生发明灯泡之后，发现他生产的灯泡灯丝总量从正极端烧断，于是进一步实验在灯泡中加入一块小金属板，给灯泡通电后将金属板连接到电表上，分别施加正电压和负电压，观察电流的情形。

对当时的科学而言，处于真空管状态下且不连接的金属在所难免，不论如何连接是不可能产生电流的。但爱迪生发现某种物质（其实就是电子）会透过金属板，从电池的负极腾空“跳”到正极，此发现称为“爱迪生效应”。这也是科学家首次质疑电流流动的方向，以及自由电子在空间中流动的现象。

## 真空管的诞生

金属之所以能导电是因为金属的自由电子较多，便于电子的相互流动，因此电子材料必须由导电性良好的材质制成。电子还有特性，带负电的电子易受到电压的吸引，所谓同性相斥、异性相吸。又从爱迪生效应得知，当加热金属物质时，活跃于质子外围的自由电子容易产生游离现象，温度高导致电子活性增强，此时若空间中有一正电压强力吸引，游离的电子就会在空间中流动。基于这几个当时已被了解的知识，弗莱明于1904年制造出第一支真空二极管，福雷斯特将真空二极管加以改良，于1907年制造出第一支真空三极管

## 真空管

三极管是最基本的真空管。

真空二极管、三极管、五极管，从字面意思代表真空管内部基本“极”的数量。真空管拥有三个最基本的极：

第一是“阴极”（Cathode，以K代表，emitter），阴极当然是阴性的，它是释放出电子流的地方，它可以是一块金属板或灯丝本身，当灯丝加热金属板时，电子就会游离出来，散布在小小的真空玻璃管内；

第二是“屏极”（Plate,以P代表,envelop），屏极连接正电压，它负责

吸引从阴极散发出来的电子，作为电子游离旅行的终点

第三是“栅极”（Grid，以G代表），从构造来看，栅极犹如一圈圈的细线圈，就如同栅栏一样，固定在阴极和屏极之间，电子流必须通过栅极才能到达屏极，在栅极之间通电压，可以控制电子的流量，它的作用就如同一个水龙头一样，具有流通与阻挡的功能。

发展史

引擎运转必须要有燃料，真空管的动作能力为电能。真空管的电极当中，最重要的应属阴极，它负责将电子释放出来，作为一切动作的基本。最早的真空管由于构造及理论简单，直接将灯丝充当阴极使用，换句话说，当灯丝点亮时，由于灯丝温度升高，电子就从灯丝释放出来，经过栅极直奔屏极。这种真空管就叫做“直热式真空管”

灯丝（filament）可以使用不同的材质制造，由于直热式三极管直接将灯丝作为阴极，因此灯丝的特性直接影响着直热式真空管的性能。基本上，真空管的灯丝主要可分三种材质构成，第一种当然是耐高温的钨丝。将高纯度的钨丝制成细丝，卷绕并装在真空管内，通电后即可发热。但钨丝必须加温到2000多度电子才能发散，因此以钨丝制成灯丝的真空管点亮时，会发出耀眼的光，同时温度高的吓人。但将钨丝点亮需要消耗较大的电力，唯一的优点是钨丝很耐用，普遍用于较大功率或长寿命的真空管中，寿命可达数万小时。

另一种灯丝采用钍钨合金，最常使用的应为氧化碱土灯丝，作法是在灯丝外，涂上一层厚厚的氧化碱土，看起来接近白灰色的物质，它只需要将灯丝加温到700度（看起来是暗红色的），即可获得足量的电子，因此工作温度低，节省电力，一般只需6.3V左右的直流就可以正常工作。

直热式电子管有一个致命的缺点，那就是阴极容易受到灯丝的温度而改变特性。当灯丝电压变动时，或以交流电供电时，阴极呈现不稳定状态

傍热式真空管

为了解决直热式真空管的灯丝问题，真空管设计者决定让灯丝与阴极分家独立，在灯丝旁套上一圈金属套筒，让灯丝直接对金属板加热，电子从金属板散发出来，这种加热方式就称为“傍热式真空管”

进一步发展

如此，真空管稳定许多了，由于金属套筒的体积与储热量远大于传统的灯丝，因此即使灯丝暂时的温度变动，甚至暂时几秒钟的停止加热，金属析的温度变化有限，这也就是为什么某些扩大机关机后，它还能响一会的主要原因。既然阴极与灯丝独立，阴极板必须由灯丝间接加热，于是灯丝再度改为钨丝，以求耐久性，并在钨丝外涂上一层白磁，

一方面绝缘，另一方面也有定型的效果。由于间接加热效果较差，阴极金属板上会涂上钽、钡或其他有利于电子发散的物质，也因此，真空管的金属极板看起来总是灰黑色的，不像正常的金属板，也由于制作组装时必须依赖手工，因此金属板上总会留下许多细小的刮痕。

### 结构

真空管具有发射电子的阴极（K）和工作时通常加上高压的阳极（屏极，P）。灯丝是一种极细的金属丝，当电流通过其中时，灯丝发热发光，以加热阴极来发射电子。栅极（G）置于阴极和屏极之间。栅极加电压是抑制通过栅极的电子量，所以能够控制阴极的屏极之间的电流

为了保持管内的真空状态，真空管内有除气剂，一般由钡、铝、钨等活泼金属合金制成，在抽出玻璃管内空气后，利用围绕玻璃管的高频电磁场使除气剂迅速升华，吸收玻璃管内的气体。在反应之后，玻璃管内壁积存银色的除气剂覆层，若把玻璃管打破或漏气时，玻璃管内壁的银色覆层会褪色，表示真空管不能被使用了。

### 工作原理

真空管具有几个极，分别为：灯丝、阴极、栅极、屏极（阳极）。当点亮灯丝时，灯丝温度逐渐升高，虽然是真空状态，但灯丝温度以辐射热的方式传导到阴极金属板，等到阴极金属板温度达到电子游离的温度时，电子就会从金属板发射出去。电子带负电，在屏极加上正电压，电子就会受到吸引而朝屏极金属板飞过去，穿过栅极而形成电子流。栅极犹如一个开关，当栅极不带电时，电子流会稳定地穿过栅极到达屏极，当在栅极上加上正电压，对于电子是吸引作用，可以增强电子流动的速度与动力；反之在栅极上加上负电压，同性相斥的原理，电子必须绕道才能到达屏极，若栅极的结构庞大，电子流可能全部被阻隔。

当栅极加正电压时，处于“放大”状态；还可处于“饱和”与“截止”状态，“饱和”即从阴极（或者叫发射极，emitter）到屏极（evelop）的电流完全导通，即栅极不带电时，相当于开关开启；“截止”即从阴极到屏极没有电流通过，相当于开关关闭，即栅极带负电。“饱和”和“截止”状态可通过调整栅极上的电压进行控制，因此真空三极管可充当开关器件，速度要比继电器快成千上万倍。

利用栅极可以轻易控制电子流的流量，将输入信号连接到栅极，并且加入适当的偏压，并且在屏极上串上一个电阻，就可达到信号放大的目的。

## 7.4 记忆力非凡的触发器

在当时，研究无线电技术或者说研究如何用电磁波来进行通信的人很多。在这些专家中，英国人埃克尔斯（William Henry Eccles）1915年出版了《无线电报手册》，1921年出版了《连续波无线电报》，现在很多电子学教材中还都称他是“发展无线电通信的先驱者”，除了研究无线电波的发送和接收技术，他还是最早注意到太阳辐射和地球外层大气会对无线电产生干扰的人之一。

和埃克尔斯一直工作的还有他的一名同事，乔丹（Frank Wilfred Jordan），他们的工作内容是研究无线电波的发射和接收。要发射电磁波，就要有振荡器，长久以来，制造振荡器的方法就是在电路中加入反馈，就像我们刚刚用非门制造的那个振荡器一样（不同之处在于，通常用于发射电磁波的振荡器产生的不是方波而是交流电那样的正弦波，只不过频率极高）。事实上，不光是那个时候，这也是我们现在采用的方法，由于老是和反馈打交道，难免会搞出一些稀奇古怪的事情来，1918年，这两个人一起发明了一种具有记忆功能的电路。

这个电路的核心是两个电子三极管，之所以让人觉得新奇，是因为它能记住你刚才对它做了什么。不过这个电路在当时却没有用处。直到过了很长时间后，因为要制造电子计算机，工程师才又发现了它，觉得这是一个好东西，不过，他们这回采用逻辑门来实现相同的功能。

这个电路上下对称，分别都是一个或门连着一个非门，特别之处在于，它们各自的输出又分别是对方的输入，换句话说，在这个电路里存在着两个反馈，如图7.14所示，在一个电路里做出两个反馈来，发明它的人还真是挺有创意

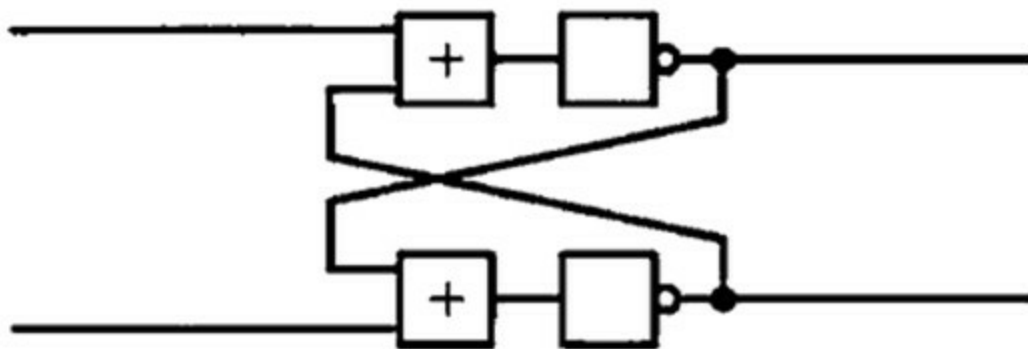


图7.14 两个或非门首尾相连形成两个反馈

这是个很奇怪的电路，上下都是一个或门连着一个非门，有两个对称的输出，可以连接两只灯泡试试

这儿说的是右边的输出，电路的左边还有两个对称的输入，很容易让人想到它是用来接开关的，因为开关可以控制输入的有无，既然有这样的想法，不妨干脆把开关和灯泡接上试试，看它有什么用，如图7.15



所示

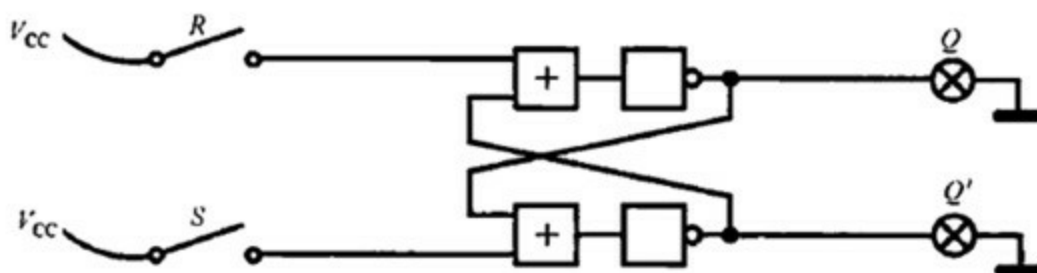


图7.15 用来验证或非门反馈功能的完整电路

我们每天都要和人交流，不管是谁，难免要提名道姓，这两个开关也有名字，一个叫R，另一个叫S，它们接在电源上，为电路中的逻辑门提供0和1.可它们为什么得叫R和S呢？后面解释。眼下，和R长在一根藤上的是灯泡Q，相应的，灯泡Q'则和开关S在一根藤上。

电路刚接好的时候，要确保两个开关都是断开的。现在要合上开关R了！

或门只要有一个输入为1，它就输出1，所以，如图7.16所示，闭合R就等于 $R=1$ ，于是不管Q以前是亮还是灭，它现在一定不会发光，即 $Q=0$

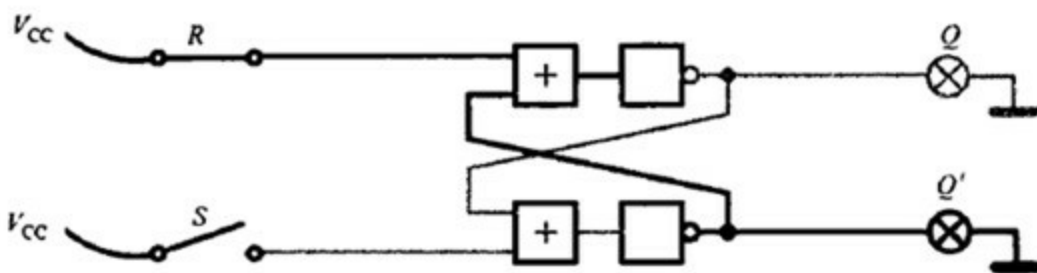


图7.16 当R闭合而S断开时，Q不亮而Q'亮

这就完了吧？没有，因为有反馈的存在， $Q=0$ 紧接着被送到下面，同时，因为S也为0，所以经或门和非门后，灯泡Q'因为被通上了电而亮起。也就是说 $Q'=1$

当然 $Q'=1$ 又被反馈到上面，但因为R已经给或门提供了1，所以Q的状态不会受到影响，整修电路就此处于稳定状态不再改变

有意思吧？很奇妙吧？断开S合上R，灯泡Q不亮，Q'亮。更神奇的是，这时，即使断开R，灯泡Q依然不亮，而灯泡Q'依然亮着

再合上R，再打开、再合上.....Q和Q'还是那样。

原因很简单，还是图7.16，因为 $Q'=1$ 被反馈到上面，所以，即使R断开，或者再合上，也不会改变或门的输出，整个电路的状态也不会发生改变。

现在，让我们把注意力转移到电路的下半部分，这一次，我们让R一直处于断开状态，将S合上，就在这一瞬间，所有的事情都颠倒了，

灯泡Q亮起，而Q'却不亮了！

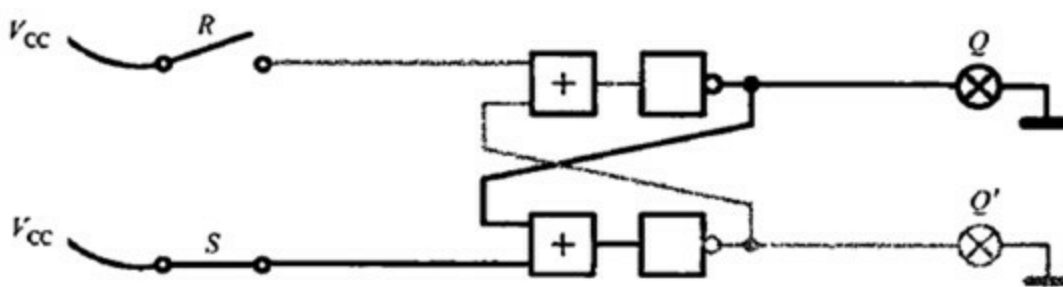


图7.17 当R断开S闭合时，Q亮而Q'不亮

由于电路是对称的，上下两部分一模一样，所以这件事情也不难理解，一旦合上开关使得 $Q=1$ 而 $Q'=0$ ，往后再怎么摆弄S（闭合或断开）都不会再影响电路的状态，换句话说，只有最开始开关的闭合是最重要的。

在这本书还没有写完的时候，我就把一部分放在了网上，大家看了之后提出了不少意见，这就是反馈。要是当初我没有这样做，那么这本书印刷出来之后，它的内容全是我自己当初的想法，但实际上我做了一件事，写了一些东西，加上读者的反馈，触发了一连串的事情，产生了完全不同的结果，也正是因为这样，我们刚刚讲的这个电路，称为触发器。

看到“触发器”这三个字，让人想到抓捕野兽的铁夹子，没错，这是一种机关，一种装置，正等着做某件事情，但就差一个外部条件

触发器英文为Flip Flop，简称FF，差不多类似于汉语中的象声词“噼里啪啦”或“噼噼啪啪”。这就是一大堆继电器在工作时所发出的声音

触发器的工作依赖于两个开关S和R，闭合一个断开另一个，总是会得到两个相反的输出Q和Q'；要是这两个开关都断开，那么，取决于Q和Q'刚才处于什么状态，它们依然保持这种状态不变。

有没有想过还有最后一种情况，要是S和R都闭合，会怎样呢？老实说，情况很不妙

如图7.18所示，闭合两个开关，将直接导致上下两个或门的输出永久为1，经过非门变换之后又都变成0，于是这两个灯泡都不亮。

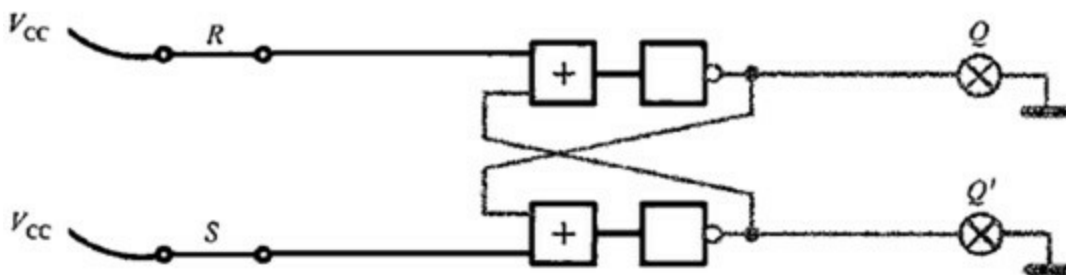


图7.18 当R和S都闭合时，Q和Q'都不亮

这是非常粗暴的做法，是暴力干涉。通常情况下，灯泡Q和Q'是互补的，配合很好，互为依托，相互制约，一个亮起来，另一个就会熄灭，能自己达到合适的稳定状态，但现在，虽然电路里依然存在反馈，但是不起作用，整修电路丧失了记忆力，差不多已经失去了理智，神经错乱了。

总结一下，这里讲的触发器，一共有4种工作状态，参见表7.1

表7.1 触发器的输出与S和R的关系

S	R	Q	Q'
0	0	不变	不变
0	1	0	1
1	0	1	0
1	1	0	0

在这里我们把R和S分别比做父母，为0表示不打孩子，为1表示打孩子；同时，把Q比做孩子对父亲的态度，把Q'比做孩子对母亲的态度。0表示不亲近，1表示亲近。那么表7.1就可以用描述如下：

爸打，妈不打，和妈亲近；

爸不打，妈打，和爸亲近；

爸不打，妈也不打，以前和谁亲近，现在仍和谁亲近；

爸打，妈也打，两个都不亲近

现在来猜猜，当这个电路刚刚通电时（R和S都是断开的），两个灯泡会是什么状态？

答案是不知道，不一定。因为这里面存在着反馈，会引发竞争，竞争的结果是肯定有一个灯亮，而另一个不亮，但到底哪个亮，哪个不亮，这个无法事先知晓，既然是竞争，那么竞争的结果只有在实际开始竞争以后才知道。竞争的结果取决于双方的实力，在这个电路中，就是零件的参数和工作速度，速度慢的一方只能在竞争中处于下风。

## 7.5 触发器的符号

像盖房子一样，触发器是构造电子计算机的重要材料，如果说逻辑门是砖石，那么触发器就是木材。为方便起见，上面介绍的触发器通常被组装成一个现成的电路，这样我们就可以不用关心它内部的结构，直接拿过来使用，如图7.19所示。

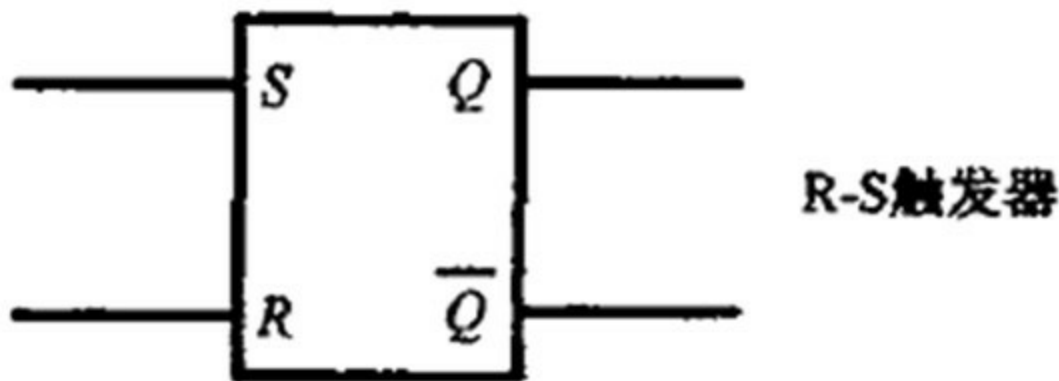


图7.19 R-S触发器符号

这是最早的，也是最基本的一种触发器，一般称它为R-S触发器。在这里，S和R不再代表开关，而Q的意思也和灯泡相去甚远，然而，无论是开关的通断还是灯泡的亮灭，代表的无非是电压或电流的有无，两种不同的表示方法，它们背后的思想是一致的。唯一不同的是以前的Q'换成了 $\bar{Q}$ 。Q和 $\bar{Q}$ 总是以相反的状态出现，Q=0，则 $\bar{Q}=1$ ；Q=1，则 $\bar{Q}=0$ 。

触发器有两个截然相反的输出，不过多数情况下我们只需要一个输出就足够。因此，一直以来把Q作为触发器的输出。结合表7.1还可以看出，在触发器正常工作的前提下，Q的输出和S的输入总是一致的，S=0则Q=0；S=1则Q=1，这意味着可以通过设置S的值，使得Q的输出和S保持一致，这就是S的由来（Set，设置）

不管Q以前是什么，比如Q=0，可以通过让S=1来使Q变成1，但，当R=1的话，Q又变回0，这等于将Q打回原形，这称为“恢复”或“复位”，R就是这么来的（Reset，复位）

## 第8章 学生时代的走马灯

每年正月十五元宵节，闹花灯，这些花灯里有一种叫做走马灯（跑马灯），制作方法是在一个大灯笼里装一个能转动的轮子，粘上各种图案，在灯笼里放一根点燃的蜡烛（或接通电源的灯泡），这时，热气升腾，轮子旋转，在光的投射下，从外面看就是一幅幅会动的图像。这种灯在宋朝就有了，当时，灯里以武将骑马的图案居多，转动时看起来好像你追我赶一样，这就是“走马灯”名称的由来。

我上学的时候制作了一种不是传统意义上的走马灯，每隔一段距离放一个灯泡，一字排开（或围成一个圆形），当它刚启动时，只有一个灯亮，紧接着，这个灯泡熄灭，与它相邻的下一个灯泡亮起，就这样，轮流发光，周而复始，循环。这样的走马灯是如何造出来的吗？这得从触发器说起

## 8.1 能保存一个比特的触发器

在埃克斯和乔丹的实验室里，触发器没有什么用途，它只是证明了电子管可以做成这么一一样东西，就像经验丰富的厨师有一天突然想到可以用蒜苗和鸡蛋做出一样新菜肴，事实上，人类的许多发明就是这样产生的。

好的东西总有用武之地，尤其是科学家和工程师喜欢翻老底子，让那些现成的发明可以“为我所用”，触发器就是一个例子。

为什么这样说呢？普通的电路以及常规的逻辑门都有一个共性，那就是输出直接依赖于输入，当输入消失的时候，输出也跟着不存在了。触发器不同，当它被触发的时候，输出会发生变化，但当输入撤销之后，输出依然能够维持。

这就是说，触发器具有记忆能力，在触发器发明若干年后，当工程师们想在计算机中保存一个比特时，他们想到了触发器。不过，触发器有两个输出，保存一个比特不需要这么多。

如图8.1所示，解决的办法是只留下一个输出 $Q$ ，而 $\bar{Q}$ 不用（把它的引脚剪掉），这样，被保存的比特可以从 $Q$ 端观察到，或者把它取走，引到别的地方使用。通过它，可以知道当前触发器保存的是什么，是0还是1

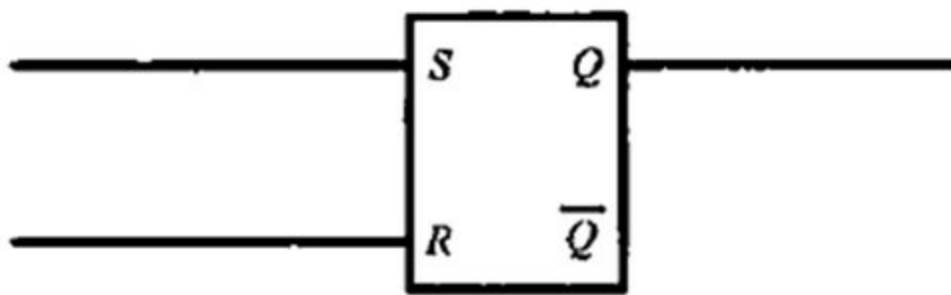


图8.1 通常把 $Q$ 作为R-S触发器的输出

不过凭什么非得是 $Q$ 而不是 $\bar{Q}$ 呢？这里面没有深奥的科学道理，只是一种约定，它们总是相反的，一个为0，一个就为1；一个为1的时候，另一个就为0。其实留下谁都可以， $Q$ 能有幸被留下，唯一的原因可能是工程师们觉得它省事儿，不需要每次书写时还要在 $Q$ 上面画一条横线。

我们的愿望是用触发器保存一个比特，一个比特只需要一根电线就可以传送，可是它有两个输入端 $S$ 和 $R$ ，而且，触发器要正常工作，离不开 $S$ 和 $R$ 的配合，要想使 $Q=0$ ， $S$ 必须为0， $R$ 必须为1；要想使 $Q=1$ ， $S$ 必

须为1而 $R=0$ 。当然在这个过程中 $\bar{Q}$ 也会发生变化，而且与Q的相反，但它的引脚被剪掉了

如何是好呢？难道为了保存一个比特，必须得用两个输入吗？解决方法是用一个非门，按图8.2所示的方法连接到触发器上

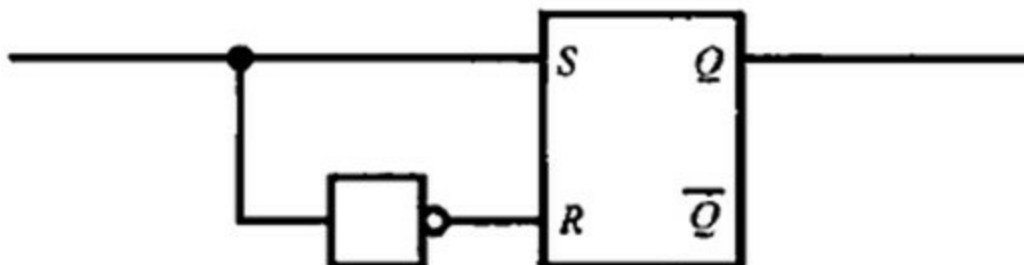


图8.2 使用非门使R与S总是相反，  
解决用触发器保存1比特需要两个输入的问题

很显然，因为要想使触发器保存一个比特，就必须使S和R以相反的方式出现，所以非门的作用就是创造这样的条件。

那么这种做法到底有没有效果呢？试试！为了看看它能不能保存一个0或1，我们在它的左边接上开关，通过闭合或断开开关，就能得到要保存的比特（0或1）。同时，触发器的输出端接了一个灯泡，灯泡的亮灭用于验证该比特是否已经被保存

如图8.3所示，电路刚接好时，开关是断开的，断开的时候相当于保存了一个比特0，这时，如图所示， $S=0, R=1$ ，触发器运作， $Q=0$ ，所以不要指望灯泡亮起来。

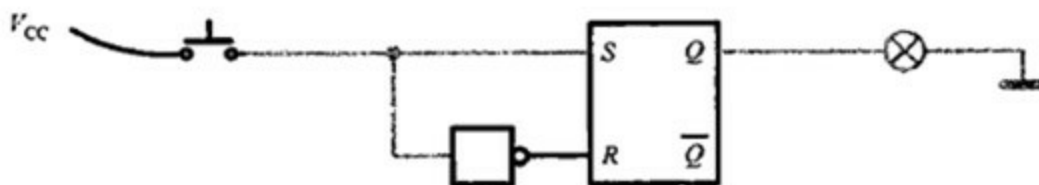


图8.3 使用触发器保存比特0的过程

灯泡灭表明目前触发器中保存的是0，现在用手按下开关，电路被接通，相当于输入1，如图8.4所示，这时， $S=1, R=0$ ，触发器保存比特1，于是灯泡亮了

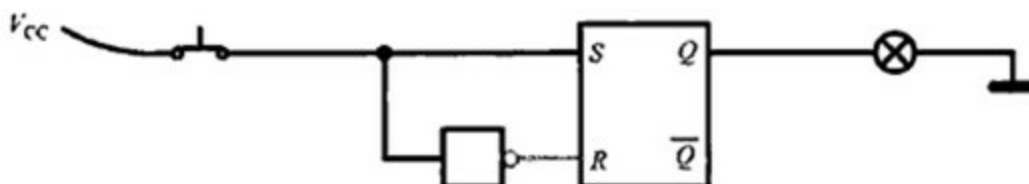


图8.4 事实证明，仅仅增加一个非门，  
并不能使R-S触发器独立地保存并维持1个比特

这是个按键开关，当松开手，自动弹起来，当开关弹开后，被保存的比特能够独立存在而不受外部的影响吗？不能，一旦按键开关弹开，灯泡立即熄灭！

原因很简单，开关弹开，相当于输入的比特是0，于是触发器又把0保存起来，灯泡自然就不亮了

看来我们没有把这个触发器设计好。给触发器安排两个门卫 – 如图8.5所示，这是两个与门，这两个门卫都归一个经理管辖，这就是控制端CP

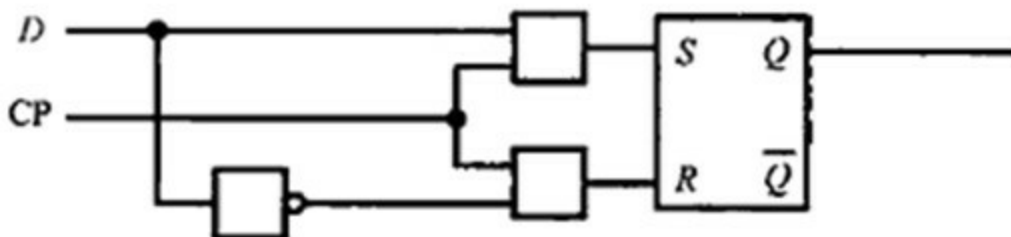


图8.5 经过改进的触发器，增加了1个控制端

这样安排电路是有用意的，而且真的是很有效。如图8.6所示，你看，通常情况下CP=0，意思是现在不想保存数据，这时，因为与门的关系，不管D上是什么S和R都为0，所以触发器保持原有的内容不变

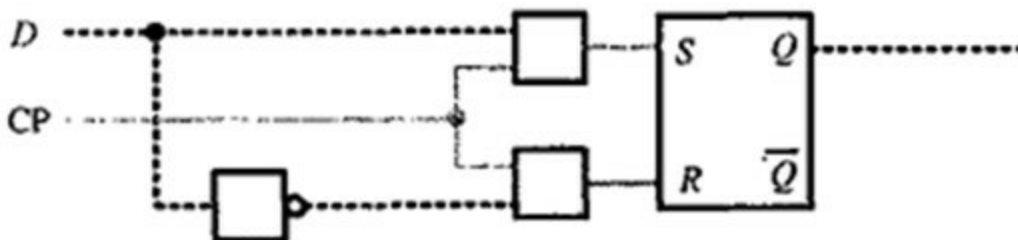


图8.6 当控制端为0时，触发器不接收D端的比特

这就是说，CP=0的另一层意思是希望触发器不被外面的数据干扰，继续保持原先保存的那个比特

在触发器前面放两个门卫（与门），不单单是保护原有的比特，它们还有更重要的任务，比如，如果D=0，而且有经理陪同前来，即CP=1,那么如图8.7所示S=0,R=1，于是0就会被保存到触发器里（Q=0）

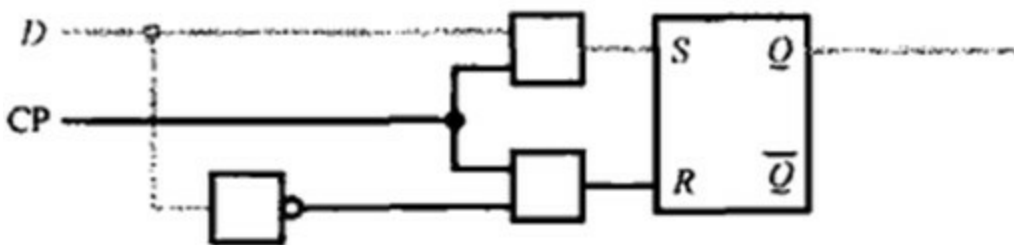


图8.7 当控制端（CP）为1时，如果D端为0，触发器的Q端为0

同样是在经理的陪同下（CP=1），要是D=1，那么S=1而R=0，于是1就被保存到触发器里（Q=1），如图8.8所示



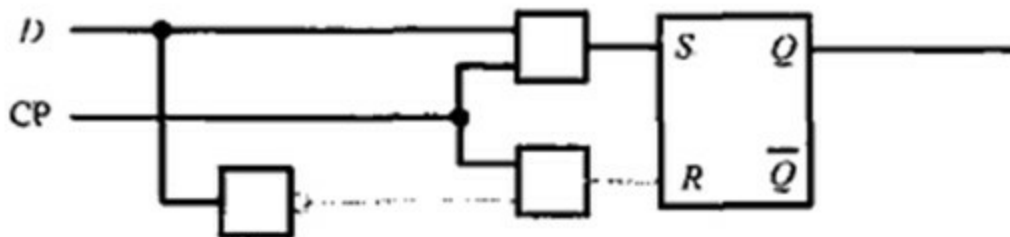


图8.8 当控制端（CP）为1时，如果D端为1，触发器的Q端为1

不管保存的是0还是1，当它成功进入触发器后，经理事情忙离开了（CP=0），于是S和R会一直为0，换句话说，没有经理的陪同，负责保卫工作的人无法确定来者是不是危险分子，谁也别想再进入触发器，触发器将一直维持刚才保存的比特不变（请再看一上图8.6）

最后，一个需要经理亲自护送才能保存比特的触发器称为D触发器，D触发器的符号如图8.9所示

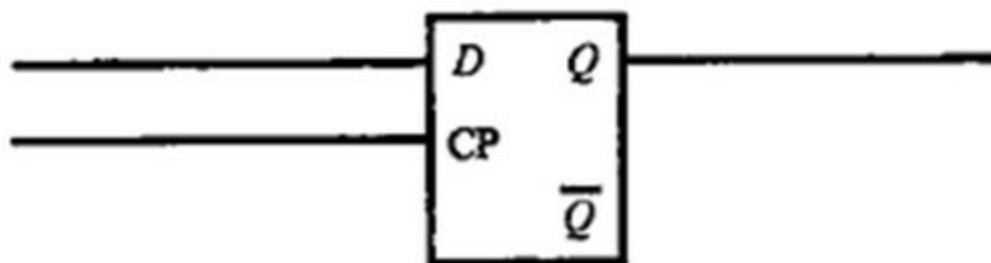


图8.9 D触发器符号

D取自英语Data的首字母，即数据。所以D触发器的名字很恰当地表明了设计它的原始目的。

## 8.2 边沿触发

对于刚接触到的D触发器来说，控制端CP就好比是触发器公司的经理，当它出现的时候，才能表明来的人是案例的。所以，也可以把CP看成是进入D触发器的通行证，不过，通行证通常是有有效期的，所以CP也不例外，它的有效期就是CP=1的时间，当CP=1时，在它的持续期间，D触发器将会卖力工作，随时都会因为外来的比特变了而触发；一旦CP=0,就意味着通行证过期了，触发器将不能保存新的比特

为了证明真的理解了上面那些话，出一道题，请你分析一下：如果从 $t_0$ 时刻开始，D端和CP端各自出现了如图8.10所示的脉冲，那么在 $t_1$ 时刻触发器里保存的是0还是1呢？

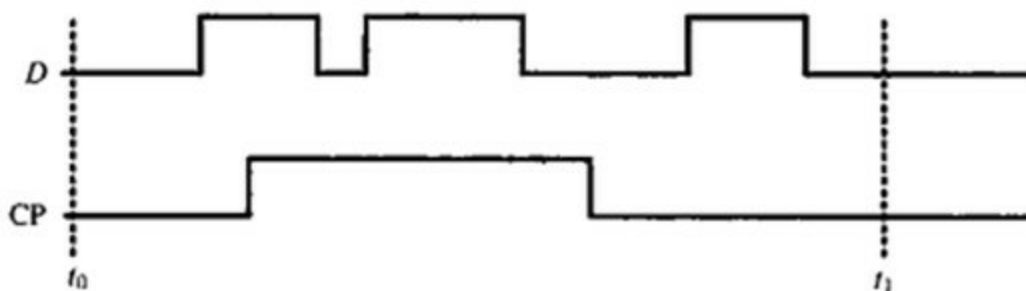


图8.10 触发器里保存的比特是0还是1，取决于CP端最后从1变为0的时刻D端的比特是0还是1

所以上图中，触发器里保存的是比特0

很明显，在CP=1期间，只要D端的比特改变了，触发器就会随时触发，所以一定要把想保存的比特放在D端，稳住，等CP从0变到1，再从1变动0之后才能松口气

这当然不错，需要的不过是细心一点。不过，要想解决懒人提出的新难题，这种触发器是不能胜任的，我们需要一种新的触发器，它只会在CP脉冲从0变成1，或者从1变成0的瞬间才会触发 – 换句话说 – 保存一个比特

这就与我们刚刚讲过的D触发器不同了，D触发器在CP为1的期间可以随时根据D端的数据触发。而现在，我们希望CP一直为0或者一直为1的期间都不会触发，只在CP从低到高或从高到低的瞬间触发

从0翻转到1，或者从1翻转到0，这个变化过程称为“跳变”或“翻转”。CP的跳变需要一个过程，可能是几纳秒，尽管非常短暂，但实际上是存在的，反映在图像上就是两个边沿，即上升沿和下降沿，如图8.11所示

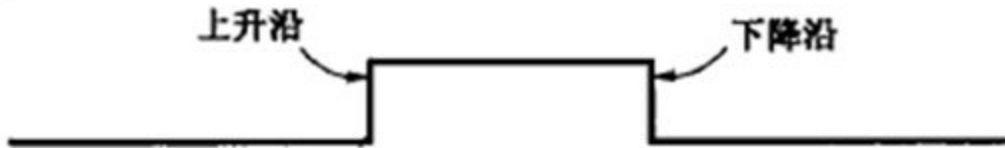


图8.11 边沿触发器的两个触发时机

下一步讲解新型的触发器“边沿”触发的D触发器，因为它只在CP脉冲的边沿触发，比如登山，迟早还得下来。所以，要说起边沿触发的触发器，实际上还分“上升沿D触发器”和“下降沿D触发器”。今天只说前一种，即上升沿D触发器

要制造一个上升沿D触发器，其实很简单，它的秘密在于可以像图8.12那样，将两个D触发器首尾相连

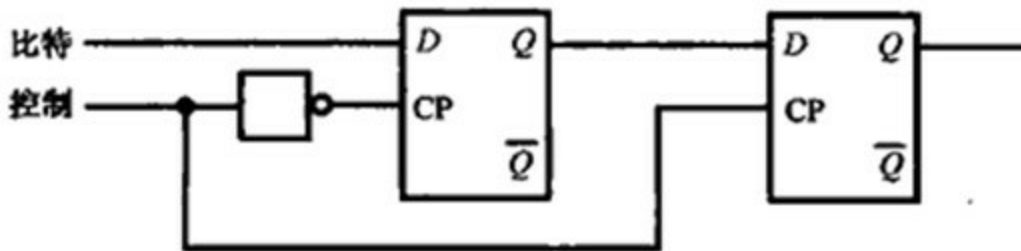


图8.12 上升沿D触发器原理

这个触发器实际上由两个D触发器首尾相连而成，前一个D触发器的输出是后一个D触发器的输入，而且，这两个触发器永远不会同时工作，因为控制脉冲是右边那个触发器的直接上级，但想要直接给左边的触发器下达命令，不行，必须通过一个非门！

如果在这个电路中保存一个比特，必须先使控制端为0，这时，左边的触发器CP=1，这样它就有了一个要保存的比特，并立即传送给右边的触发器，但遗憾的是右边的触发器不工作

现在，如果控制端从0跳变到1，左边的触发器不再接受任何比特，右边的触发器活跃起来，把左边触发器的输出保存起来。换句话说，直到这个时候，比特才算是被保存起来了

此后，如果控制端从1变回0，即下降沿，左边的触发器苏醒过来，但右边的触发器却开始休眠，但它仍有能力维持原先的输出不变，这就是说，控制脉冲的下降沿不会改变这个触发器的内容

上面这个二合一的触发器不管控制端是0、1还是从1到0的下降沿，它都不能保存比特，除非一种情况，那就是从0到1的跳变，即上升沿，为了便于表示，上升沿D触发器的符号如图8.13所示。注意，和普通的D触发器不同，它的控制端CP旁边有一个三角形，表明它是边沿触发的

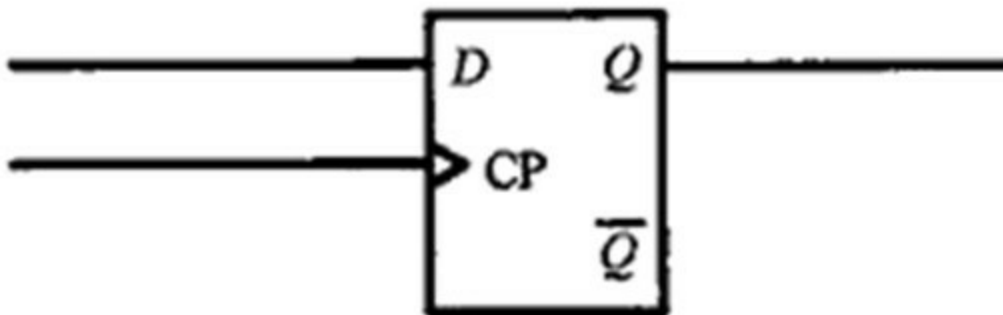


图8.13 上升沿D触发器的符号

为了证明你确实已经理解了上升沿D触发器的工作原理，这儿有一个小小的考试题：回到图8.10，请问，同样在 $t_1$ 时刻，上升沿D触发器里保存的是什么？是0还是1？

## 8.3 揭开走马灯之谜

讲了这么多，现在到了揭开走马灯神秘面纱的时候了。很幸运，我们只需要一些触发器就可以解决大部分问题。不过，我们需要的不是传统的触发器，比如R-S触发器，而只能用上升沿D触发器

做任何事情都需要一个过程，把一只老鼠放进管道，要过一会它才能从管道的另一头钻出来。上升沿D触发器只在CP脉冲的上升沿触发，比特从输入端经过一个小小的延迟后才能稳定地出现在输出端Q上。延迟的时间或长或短取决于制造触发器所使用的材料，如果用的是继电器，那么延迟以秒为单位，大约是零点几秒到1秒，因为它不单纯是电子的更是机械的；如果用的是电子管或更好的材料（这些东西后面介绍），那么，这个延迟可以缩短到几纳秒（1等于十亿纳秒）

基于这一点，可以把若干个上升沿D触发器首尾相连，并把它们的CP端连在一起，让它们可以在同一时间触发，如图8.14所示

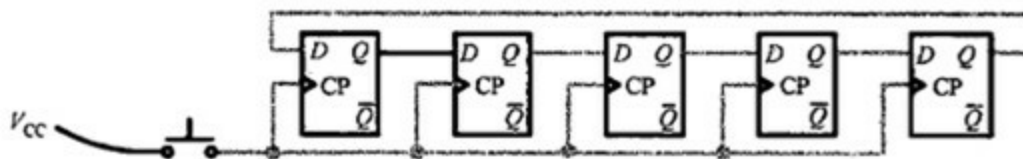


图8.14 几个首尾相连的触发器使用同一个控制端，能同时触发

如图所示，我们想办法让最左边的触发器保存比特1，而其他的触发器都保存0，通常情况下，按键开关是断开的，所有的触发器都不工作，因为它们的CP都为0。一旦开关被按下，在电路接通的瞬间，所有的触发器都会看到一个上升沿，于是它们的第一反应是将前一个触发器的输出保存起来，一个小小的延迟后，它们将前一个触发器的输出出现在各自的输出端Q上。但是由于上升沿已经过去，各个触发器就此停下来，不过此时会发现，第一个触发器的输出Q已经不再为1，取而代之的是第二个触发器

这就是说，如果不停地按开关，这个比特1就会在触发器间顺序传递，从左到右，最后又回到左边，循环往复

现在，要是每个触发器在把输出送到下一个触发器的同时，还连着一个灯泡，这不就是我们想要的走马灯吗？

是的，完全正确，但问题是，当把这样一个走马灯制作成功时，谁来不停地按按键开关呢？这太麻烦了

上一章我们已经用非门制造了振荡器，非门振荡器的输出和用手反复按开关所产生的效果是一样的，既然如此，就不用人工来反复按开关了，接一个即可，如图8.15所示

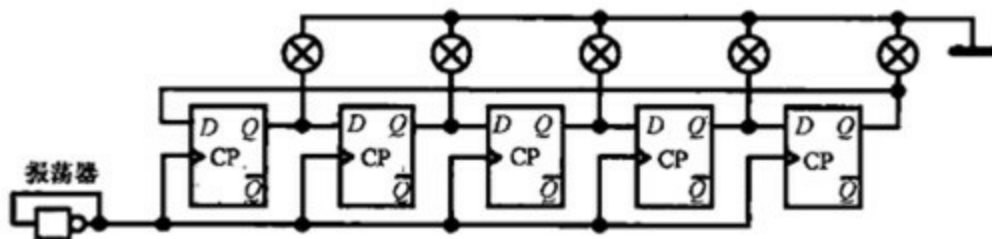


图8.15 采用上升沿D触发器和振荡器的走马灯电路

这个走马灯的速度是可调的，如果振荡器的振荡频率很高的话，灯泡亮灭之间的变换就很快，反之则慢。那么，为什么制作这样一个走马灯非得使用上升沿D触发器吗？

振荡器所产生的是一连串交替变化的0和1，不像上升沿D触发器，普通的D触发器可以连续触发，只要它的CP为1，如果走马灯使用普通的D触发器来制造，在每个CP=1的持续时间，它们可能会连续触发，从而使得它们之间的比特1连续穿越好几个触发器，直到CP=0，这样的话，走马灯也就成了飞马灯

不把振荡器和灯泡算在内，一个走马灯电路通常称为循环移位寄存器，所谓“寄存”的意思是临时存放，就像火车站旁边的物品寄存处，触发器随时会根据需要而保存新的比特（如果希望得到一个不变的0和1不需要使用触发器），仿佛这些比特都是发射星云寄存在触发器里，当若干个触发器组合在一起，可以同时保存许多比特时，就称为寄存器

不过，不要以为世界上只有循环移位寄存器，它只是另一种寄存器——移位寄存器的特例。移位寄存器可以设计用来保存多个比特，并将这些比特顺序左移或右移。当用加法机做加法时，加数和被加数的所有比特都是同时进入加法机的，这称为“并行”

但在另一些场合，为了迁就已有的电路和设施（毕竟在制作这些东西时花了钱，只要还能凑合，没有人愿意多花钱），二进制数（或二进制比特串）需要按顺序拆开，然后一个比特一个比特地分别传送，到达目的地后再按原来的次序组装起来，这称为“串行”。在这方面，一个典型的例子就是通过电话线上网，或者向U盘复制资料。在这种情况下，所有的信息都是通过电话线或数据线一个比特一个比特传送的

为了将原本“并行”的数据“串行”地发送出去，就要用到移位寄存器，很显然，循环移位寄存器只不过是一个首尾相连的移位寄存器，我们讲了这么多，唯一没有说明的是如何把要移位的数据保存到移位寄存器（的每个触发器）里。这的确很遗憾，因为要做到这一点，需要添加额外的电路，而这超出了本书的范围，所以只能留给大家自行探索

## 8.4 这个触发器很奇怪

一个走马灯的确可以增加节日气氛。在本章的最后，我们来见识一个新的触发器，老实说，这可能是迄今为止你见过的最古怪的触发器！实际上，这是一个稍加改造的上升沿D触发器，如图8.16所示

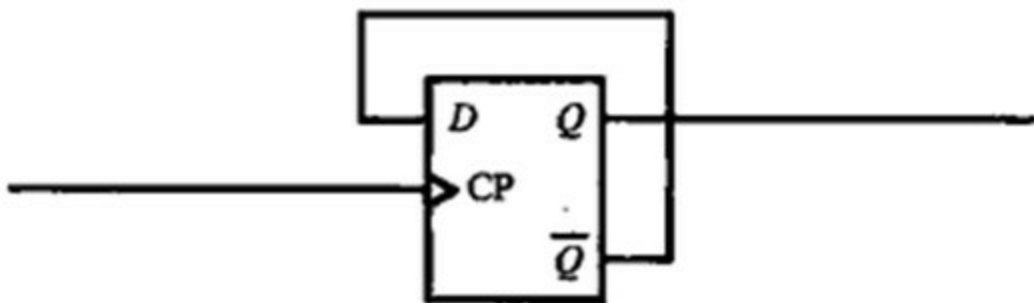


图8.16 首尾相连的上升沿D触发器

改装后的触发器显得很奇怪 – 很古怪，它的输出 $\bar{Q}$ 又被送到输入端D，看起来像是一个反馈。那么它是如何工作的呢？

为了清楚它的工作原理，我们需要另外两个道具：开关和灯泡。开关用于生成控制脉冲，而灯泡可以使我们观察到开关对灯泡亮灭会产生什么影响

发图8.17所示，我们知道，触发器有两个相反的输出Q和 $\bar{Q}$ ，现在假设Q=1而 $\bar{Q}=0$ ，灯泡是亮着的

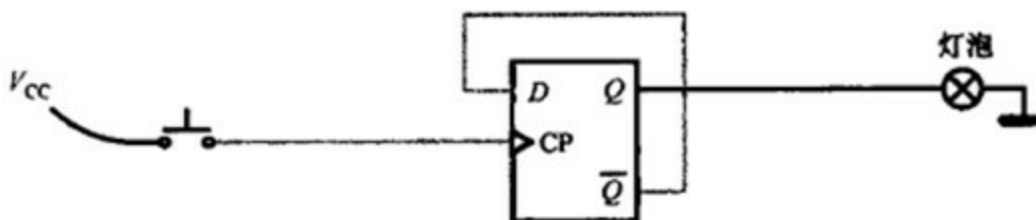


图8.17 事实证明，反复按动开关，灯泡就会在亮灭之间交替变化。只要开关是断开的，CP端就看不到上升沿，触发器也不会工作，所以从 $\bar{Q}$ 到D上的反馈在触发器内部被阻断

现在，因为 $\bar{Q}=0$ ，所以D也为0，如果按下开关，在它的上升沿，触发器运作，将D上的比特0保存，结果是导致Q和 $\bar{Q}$ 同时翻转，Q=0而 $\bar{Q}=1$ ，灯泡熄灭

触发器更像一根管道，D上的比特被保存后，需要经历一个小小的延迟，才会使Q和 $\bar{Q}$ 产生变化。实际上，当Q和 $\bar{Q}$ 翻转后，CP的上升沿已经过去，在下一个上升沿到来之前，从 $\bar{Q}$ 到D上的反馈又一次被阻断。

同时可以分析，在又一次按下开关后， $Q=1$ 而 $\bar{Q}=0$ ，灯泡又亮了，换句话说，它就是一个反复开关，当按一下开关，灯亮了；再按一下，灯灭了；再按一下，灯又亮了，再按一下，灯又灭了.....

尽管只是简单的改装了一下，用一个普通的上升沿D触发器，在它的D和 $\bar{Q}$ 之间连了一根电线，但它仍然有资格被认为是一种新型的触发器，至于它的名字，有很多种叫法，有一种叫法是T触发器，T是英语单词Toggle首字母，意思是“反复”



## 第9章 计算机时候的开路先锋

在前面我们讲了各种各样的触发器。触发器可以由与、或、非门构成，如果说这些最基本的与、或、非门是混凝土、沙子和钢筋的话，触发器就是盖房子要用到的大梁和预制板

触发器当然不能用来盖房子，而它也并非真正的预制板。我打这个比方，其实是想说，这是一种很重要的东西，电子表、电子钟、手机，甚至这本书要讲的主角－电子计算机都离不开它

所以讲触发器只是一个铺垫，我们真正要讲的是计算机为什么会自动工作（计算），这种“自动”本质上是怎么发生的。而且这本书前半部分基本上都是铺垫，该切入正题了

不过仔细检查发现这个垫子还少一块，而且缺少一些点缀，所以在这一章，让我们把前期工作完成，再开始进入正题。

## 9.1 纯电子化的计算时代

我曾经看到过一篇文章，说最早的计算机是电子管的，这倒没错，事实的确如此。不过，在他们的字里行间的意思是因为要发明电子计算机，所以才有了电子管，这真是奇谈怪论

世界上第一只电子二极管发明于1904年，1906年，福雷斯特发明了三极管。电子二极管和三极管的发明不是因为要制造计算机，相反，它们被广泛应用于电话、电报和无线电通信。原因很简单，因为现代的电子计算机还没有走过理论准备阶段。尽管1918年埃克斯和乔丹发明了世界上第一个触发器装置，但他们尚不知道这东西有什么用处。

直到1936年，情况也没有太大的改观，唯一的例外是香农发表了她的论文《继电器和开关电路的符号化分析》。这篇论文里打开了进入数字电路王国的第一道关口，那就是如何根据我们想要得到的结果来构造一个电路。

香农的工作是基础性的，特别是对于制造现代的电子计算机，具有讽刺性的是，数字电路的第一个应用并不是电子计算机，而是被用作电话交换。事实上，这也正是香农本来的目标。我们知道，每一部电话都应该和其他任何一部电话相连，这是保证它们互相能够通话的必要条件。但，不可能真的把全国、全世界的电话都物理上一对一连接起来，这样做成本高的无法实现。解决之道是让它们共用一些有限的线路，只有当某两个电话要互相通话时，才临时接通物理线路。

最早的时候，使一些电话互相接通的工作是由人工完成的，这些负责接线的人称为接线员。接线员通常是女性，因为她们有耐心。这样，远程通话可能需要多次转接。

电话接线的岗位能提供大量的就业机会，但科学家想的是“能让打电话的麻烦程度降到最低吗？”这样，当开关电路的理论建立起来之后，第一个受益的就是电话交换，当一部电话拨号时，交换电路就会产生一个输出，使某个继电器吸合，从而将两部电话接通。本质上，我们现在的电话交换机也是这么工作的。

数学和逻辑电路的发展吸引了一批年轻的科学家，使他们看到了从事计算机研究的前途。传统上，所有的计算机都是机械的，比如算盘和齿轮。机械设备并不是不好，如起重机、挖掘机.....但要用机械来制造计算机、进行数学计算，就显得很笨拙：精度不高、速度慢、操作起来很麻烦

在那个时代，科学家们知道继电器可以用来制造逻辑门。那么可以用逻辑门来搭建各种各样的逻辑电路，包括那些用于解决数学问题的运

算电路。

用继电器制造的运算电路是一个名副其实的怪胎。为什么这么说呢？原因在于，它的工作需要电流的驱动，但它的吸合和释放却是一个机械过程，换句话说，它一半是电子的，一半是机械的。除此之外，它的工作速度也不令人满意，当然可以用它来制造触发器，但这些产品不能在较高的频率下工作，因为它的触发过程不会在瞬间完成。

计算机应当摆脱机械，包括笨拙的继电器，实现完全的电子化的运算，这对那些电子计算机的先驱们来说，应该是很自然的想法。纯电子化的运算很奇妙，两股代表着不同数值的电流在某个装置内汇合，互相影响，变成另一股合适的电流，这就是计算结果。最重要的是在这个过程中没有机械的影子，看不到继电器衔铁的吸合与释放，也听不到噼啪声，第一个做出这项变革的，是约翰·文森特·阿塔纳索夫

阿塔纳索夫1903年生于美国，父亲是来自保加利亚的移民，母亲是数学老师。从1936年起，他在艾奥瓦州立大学的物理系任副主任。艾奥瓦州位于美国中部

1937年，在阿塔纳索夫34岁那年的冬天，他在这里找到了使计算机实现纯电子化运算的答案。这一年，实际上就是即将结束的机械时代与即将到来的数字时代之间的交接点。从这一点来说，1937年应该被称为电子计算机元年。

阿塔纳索夫的核心思想是使用二进制，并采用电子管来制作进行加减乘除所需要的逻辑电路。电子管是新材料，继电器能做的事情它也能做。比如，可以利用电子三极管的栅极来控制阴极和阳极之间的通断，这就相当于一个逻辑上的非门。电子管还有一些继电器不具备的优势，它是纯电子的，开关速度要比继电器快成千上万倍

对于早期的计算机来说，电子管是好东西。遗憾的是因为比灯泡复杂了很多，所以价格自然也很贵，即使是在它发明二十年后1937年，视质量的好坏，一只电子管也要卖几美元甚至十几美元

这还不是最主要的，任何科学家都需要完备的理论作为支撑，计算机也不例外，尽管在那个时候电子管无疑是最好的材料，但计算机理论准备却刚刚开始。20世纪20年代到40年代，电子管在各行各业中的应用如日中天，特别是无线电广播在无数无线电爱好者的努力下得到迅速发展，收音机开始潮水般涌上市场，巨大的需求刺激着电子管的生产和销售。据说当时一个规模比较大的企业每年生产的电子管数量在百万个以上

所以，电子管的这种兴旺和计算机没有关系，计算机一直在按自己的步调慢慢发展，但就在这个过程中，另一种比电子管更好的东西出现了

## 9.2 晶体管时代

撇开制造计算机不论，即使对于其他行业来说，使用电子管也有一些不便之处，电子管体积太大，数量一多就比较占地方，还有，要让它老老实实干活，必须依靠灯丝把阴极烧热，这灯丝跟灯泡的灯丝一样，消耗电，发出光和热，成千上万的电子管加起来还有它们消耗的电量花费太高了

说到灯丝，不得不提的是，因为从灯丝发热到把电子管阴极加热到能发射电子，需要一段时间，称为预热。想想老式电视机，因为显示装置和电子管有些类似，是通过阴极发射电子轰击荧光屏来显示图像的，所以每次打开电视电源后，需要一小会儿才能看到图像，就是这个原因。

最后，电子管怕振动，过分振动会损坏灯丝，在灼热的时候很容易因为振动而断掉

所以很多科学家忙着新的发明，准备革电子管的命。在科学家史上往常那样误打误撞不同，这是一次目标非常明确的行动，就是要用新的材料来取代电子管。成功的人是肖克利。

肖克利1910年生于英国伦敦，父母都是地质学家，3岁的时候，他跟随父母到了美国。他的父亲威廉·赫尔曼·肖克利是采矿工程师，常年在矿区生活。他的母亲梅·布拉德福·肖克利，是20世纪早期少数从事地质学工作的女性之一，有雄厚的岩石和矿物知识，受她的影响，肖克利10岁时候就成了一个岩石迷，后来他在麻省理工大学读固体物理学，固体物理学说白了就是研究固体物质材料中的电子。他获得了博士学位，留校任教。

1936年，肖克利从麻省理工大学来到著名的贝尔实验室，由于性格方面的原因，他在这里的名声并不是很好，据说他对待同事态度粗暴，到后来，当他从走廊里经过时，大家都会避开他。但他有刮目相看的聪明才智，1947年，他和两位同事发明了晶体管，1948年申请专利，1956年，他们三个人共同获得了诺贝尔物理学奖。

晶体管是电子管更好的替代品，被称为“20世纪最重要的发明”。说到晶体，顾名思义，就是那些有光泽的东西，这个词和汉语里的其他词汇，如结晶、晶莹有一定的关联。不过晶体不一定非得是白白亮亮的，像食盐，有些东西虽然灰头土面，如石墨，但在光线的照射下依然“晶光闪闪”。

在物质中，晶体只占了一部分，和其他物质相比，晶体总是有规则的形状，即使打碎了也是这样。另外，它们有固定的熔点，如铁，就算

把它烧的红彤彤的，但温度达不到1535摄氏度，它是不会熔化的，还是硬梆梆的。相比之下，尽管有些东西看起来也是亮晶晶的，但它们并非晶体，如玻璃。和铁不同，玻璃谈不上熔点，达到一定温度开始变软，直至熔化。

以肉眼来分辨晶体和非晶体不是很可靠，但在微观上，也就是在原子层面，它们之间的差别却很清楚。组成晶体的原子排列的很规则，而且原子之间的关系也很稳固，这都是电子的功劳。正是因为这样，所以肉眼看来它们有固定的形状，要是温度不够高，是不足以破坏原子之间的那种稳定关系的。

所有的金属都是晶体，其他物质，不管它们导不导电，很多也是晶体，比如面碱、糖、味精、人的牙齿和骨头等等。当然钻石也是晶体。

晶体那么多，却都不是本章的主角，真正的主角是我们相对来说不大注意的硅和锗。

硅在这个地球上含量丰富到处都是，据测定，硅占了地表岩石的四分之一，硅在1822年由瑞典化学家发现。锗1886年在德国被发现。

但制造晶体需要纯净的硅和锗，也就是在一块硅和锗里基本不含其他原子。遗憾的是硅在自然界里几乎都是以化合物的形式存在的，比如二氧化硅。化合物就是几种不同的原子或分子互相结合而形成的另外一种截然不同的物质。通常，纯正的硅是一种暗色的，但有点儿光泽的固体，当温度很高时，硅和氧气结合形成二氧化硅，也就是沙子和岩石。地球上的沙子和岩石应该就是这么来的，由此可见地球刚刚形成时表面温度是很高的。

要得到纯净的硅，就得从硅的化合物中把其他原子赶走，比如二氧化硅中的氧。不过，这可不像把土豆上的泥洗掉那么简单，需要很高的温度，经过好多道复杂的工序，既费事又费钱。即使是这样，也不可能得到完全纯净的硅，换句话说，它里面还留有少数其他原子。毕竟从二氧化硅中把氧去掉不像从大米里拣出石子。在这方面，能达到的最高成就就是把硅的纯度提高到99.9999999999%，要是知道豌豆那么大的一块硅里面有多少个硅原子的话，差不多就能算出还有几个其他原子混在里面。

纯净的硅叫本征硅，意思可能是说这种人工提炼的硅才具有硅的本质特征，因为自然办里的硅也叫“硅”，但它们都是化合物，不过，“本征”这个词好像仅仅用来指硅、锗这类物质，极少用在别的东西上。你可以说“这头猪是纯种的”，要是你说“这是一头本征猪”会有多么别扭。

和金属相比，硅和锗的导电性很差，不过比顽固的绝缘体活跃些。所以，它们被称为“半导体”

取一块纯净本征的半导体，如图9.1所示的那样，在一边渗上硼，

另一边掺上磷，然后分别引出两根导线，会发生一些古怪的事情：不但这块半导体的导电性能获得了很大的改善，而且像电子二极管一样，具有单向导电性。

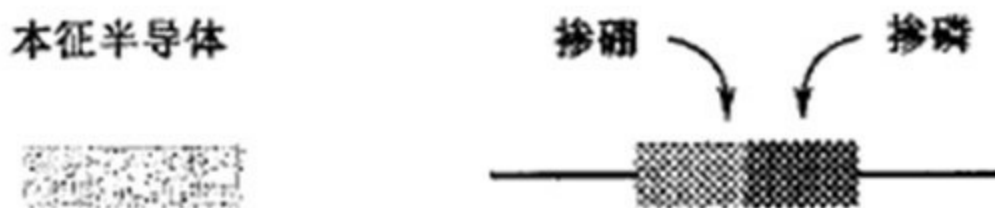


图9.1 本征半导体的掺杂方法

掺杂需要在能有杂质气体的高温炉里进行，当硅和锗处于熔融状态时，杂质气体便能渗透到里面，并产生一些奇特的物理过程。在掺杂的过程中，像氧气这类的东西是不受欢迎的，必须杜绝。否则一不留神就掺成二氧化硅了，而二氧化硅到处都是，用不着这么费劲地去制造。

不是所有的东西都可以通过掺杂而产生单向导电性。比如，要是采用相同的方法给食盐掺杂，就不会产生奇迹。所以归根到底这属于半导体物质的特有性质。从晶体的性质到半导体的性质，再到掺杂时半导体内部会发生怎样的物理变化，这一切不是这里能说清楚的。

因为硅和锗是晶体，所以这个具有单向导电性的装置就叫晶体二极管。晶体二极管根据用途分为很多种，形状也五花八门，不尽相同。小的像颗米粒，大些像个大蒜。说到这里，也许大家对它感到既陌生又遥远，其实它一直在你身边。在发明晶体二极管没多久，人们就发现如果用半导体中掺入砷、镓等原子，制作出来的晶体二极管就会发光，称为发光二极管（LED），要是进行一些特殊处理，还可以控制光的颜色，从那以后，发光二极管被越来越广泛地用到所有可能的地方，在你使用的计算机上，显示器、电源指示灯是发光二极管；电饭锅、微波炉、电视机上的指示灯还是发光二极管。在2008年北京奥运会上，据统计，所使用的发光二极管总数可达好几十万。发光二极管体积小，不需要高热的灯丝就能发光，更重要的是它耗电量非常小，仅仅这一个优点就使它具备广泛推广应用的价值。

和电子二极管一样，晶体二极管只具有单向导电性，不具备放大作用。不过，在接着探索了一段时间后，如图9.2所示，在一块本征半导体的两边掺上硼，在中间掺上磷（中间这个区域一般做得比较薄，大约1微米到十几微米，而且掺得很少。微米记为 $\mu\text{m}$ ，1米=1000000微米，即1米等于1百万微米），这样就发明了一种新型的半导体材料。

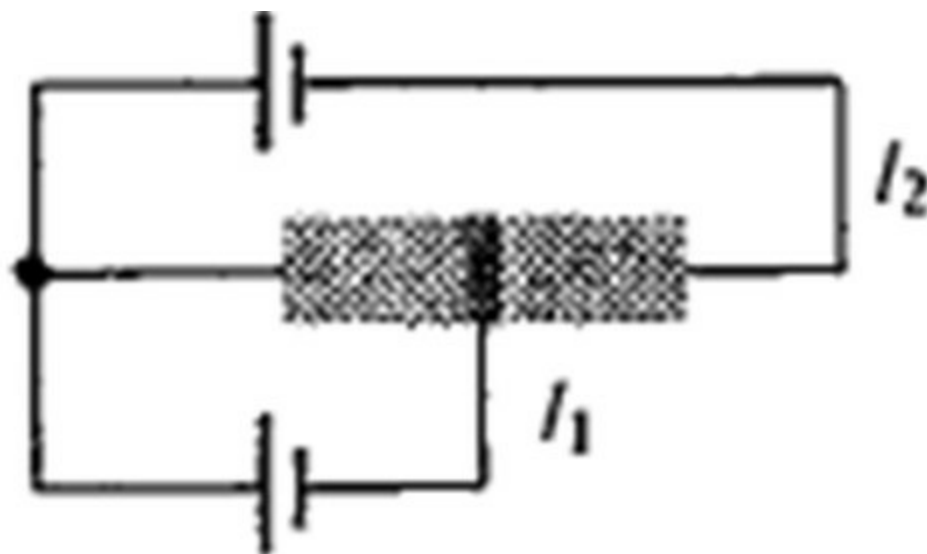


图9.2 晶体三极管工作原理示意图

看上去我们是在做两个背靠背紧挨在一起的晶体二极管。这种算法当然没有错，但不是它真正的价值所在。现在，像图中所示的那样，为这块半导体的三个部分通电，这时，你会惊讶地发现，只要电流 $I_1$ 发生一点变化，电流 $I_2$ 就会大幅度地跟着变化。也就是说，这个新的半导体材料像电子三极管一样，具有放大作用。相应地，它被称为晶体三极管和晶体二极管一样，晶体三极管也是种类繁多，长什么样的都有。图9.3左边显示的就是其中的一个种类



图9.3 常见的几种半导体器件外观

图9.3还显示了常见的晶体二极管和发光二极管。但不要按图索骥，因为它们有各种各样的型号，形状大小也迥然不同。

不像电子管，晶体管可以做得很小，很轻巧，不需要很高的电压就能工作，更不需要一个灯丝来加热。可想而知，当晶体管大量生产之后，会刮起怎样的普及风暴（新生事物的成长过程通常不会一帆风顺。晶体管出现的时候，由于其工艺、性能方面还有待提高，加上输出功率很小，业界都不看好，有人甚至断言“晶体管不可能取代电子管”）

实际上，在了解了晶体管的本质后，人们多多少少都会有些特别的想法，比如，晶体管的工作原理和一块硅的大小实际没有关系，可以将晶体管做得很小，也许可以做到连肉眼也看不到，但丝毫不影响它的单向导电性，也照样可以放大信号。因为即使一块硅小到连肉眼都看不到，它还是一块硅，依然有数不清的硅原子。本来晶体管是可以做得很小的，但是要用在电路上，它必须有一个外壳，以防止损坏，还得引出导线以方便连接 – 所有这一切都使得晶体管在做成实际的产品之后显得有些大。

和电子管一样，晶体管也是制作逻辑门，乃至各种触发器的好材料。而且，使用晶体管，可以更省电、体积更小、且轻巧耐用

如何使用电子管和晶体管来制造逻辑门，这不是本书的话题。希望大家在看了这本书之后，可以自己翻翻相关书籍

在发明了晶体管之后，肖克利一心想要发大财，他离开了贝尔实验室，去制造晶体管。

由于发明了晶体管，而且获得了诺贝尔物理学奖，肖克利被认为是当世大贤，受人仰慕。一听说他出来单干，而且正需要人才，许多年轻人从四面八方赶来，准备和偶像一起建立丰功伟业。但时间一长，大家才明白肖克利虽然在智力上超群，但为人傲慢，不懂市场和管理，很难与人相处，就像有人所说，他是“一个天才，又是一个十足的废物”。

最开始的时候这还能忍受，到后来分歧越来越大，1957年7月，肖克利发现有一些做实验用的金线不见了，盛怒之下打算让他的八个学生接受测谎仪的测试。在对不再抱任何幻想后，这八个他最得意的学生最终选择了集体出逃。

至于肖克利本人，在经历了毫无建树的岁月后，1963年，他离开自己的公司去大学做了一名教授，70年代，他忽然对人种和优生学发生了兴趣，在做了一番所谓的研究后，公然宣称并不是所有人上遗传上都处于同等水平，他们也不是在同等的基礎上进化。更惊人的是他还发表论文，宣称黑人的智商要比白人低20%，愤怒的黑人学生起来抗议，在校园里焚烧他的肖像。



## 9.3 新材料带动技术进步

人类从来不缺乏梦想，但这些梦想能不能实现却要看有没有合适的工具和材料，人们经常说要“现实一点”，就是这个意思。前不久我回了一趟家乡，对此深有感触。从长春到湖北十堰，相距四千多里，要在古代，骑马最快也得几个星期吧；现在坐火车30个小时；如果是高铁只需要10几个小时。这就是新技术、新材料带来的进步。

电子管和晶体管的应用也是这样，在没有这两样东西时，我们从来不曾听过收音机，也没有看过电视和电影，哪怕是幻灯片也没看过，电子计算机就更别提了。但，当电子管和晶体管发明后，整个世界很快变得丰富多彩了

要清楚电子管和晶体管到底有多少种用途是不可能的，因为我们总是会产生更多的需求。比如现在，在本章的最后，我们就有一个特别的需求，事情是这样的，我们想发明一个装置，来帮助制药厂自动统计药丸的数量。

注意，这不是个普通的例子，名义上我们是在给制药厂帮忙，但我们要发明的东西电子计算机也同样用得着，而且毫不夸张地说，电子计算机少了它就不行。

在药厂车间里，药丸生产出来后，要通过一个有孔的装置掉下来，然后集中装瓶。如何统计药丸的数量？制药厂采用的是光电技术，如图9.4所示，在药粒下落部位的一边，是一个发光装置，可以定向发射一束光线

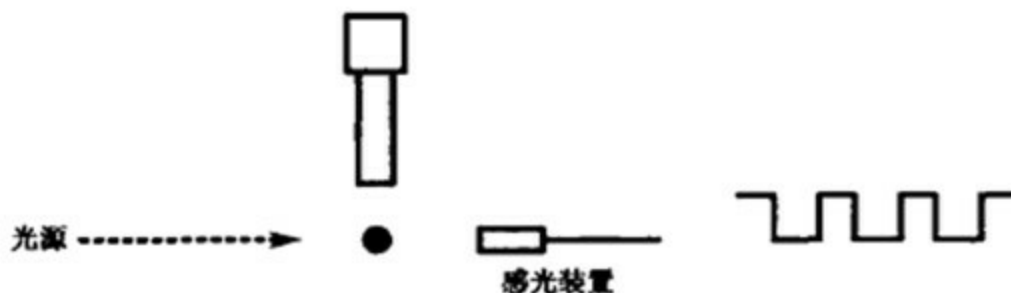


图9.4 假想的药丸统计原理

在另一边，是一个光电转换器，这是一种半导体器件，现在用得非常广泛，在光线的照射下可以产生电流。后面还要讲到光电转换器。

平时在没有药丸掉落的时候，光线无阻挡地照射到光电转换器上，从而产生电流，反之，如果有药丸掉下来，光线被挡住，光电转换器不再有电流产生。从而使光电转换器产生断断续续的电流输出，经过修整后，就是一连串的方波脉冲。现在要做的就是统计脉冲的个数

脉冲的个数就是落下的药丸的数量，而要统计脉冲的个数，前面讲过的反复触发器的知识，如图9.5所示

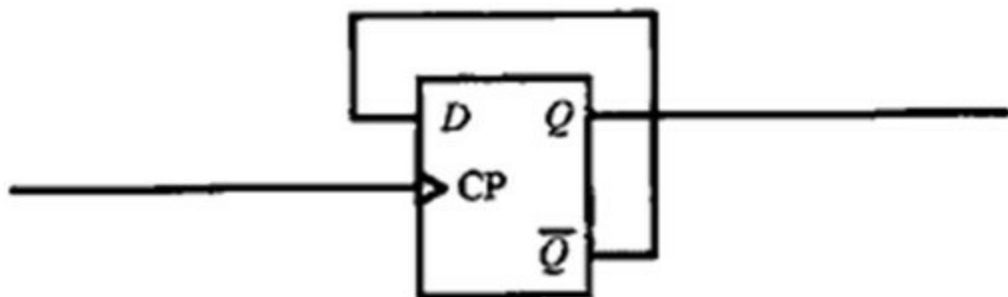


图9.5 反复触发器示意图

每当一个脉冲到达CP端的时候，这个触发器输出Q就会发生翻转，如果原先是0则变成1；如果原来是1则变成0，而 $\bar{Q}$ 也是这样，只不过它与Q的值正好相反。

为了在后面讨论问题的时候画起来方便，我们把反复触发器用图9.6那样的符号表示，看得出，我们隐藏（省略）了输出端 $\bar{Q}$ 和输入端D之间的连接

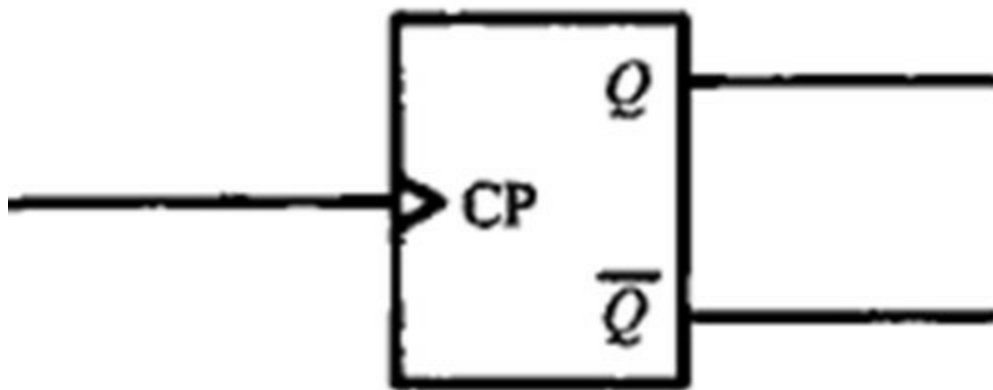


图9.6 反复触发器的符号

取决于要统计的药丸数量有多少，可能需要许多这样的触发器，并将它们首尾相连，这样就造出一种可以计数的东西，也就是计数器。如图9.7所示

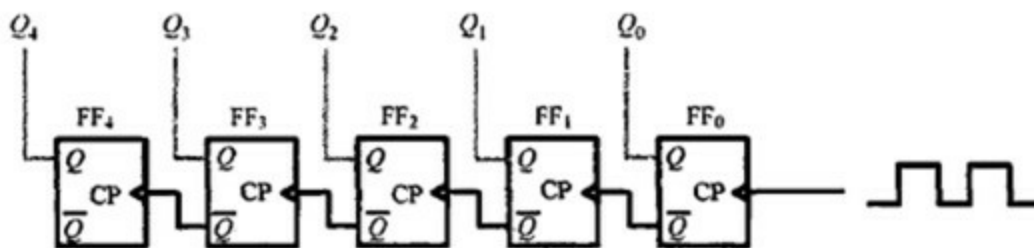


图9.7 5个反复触发器构成的计数器

这个计数器用了5只反复触发器。注意每个触发器，它们和以前的视角不一样，这完全是为了大家看起来方便。

很明显，从右向左看，每个触发器的输出端 $\bar{Q}$ 都连着下一个触发器的控制端CP，在这个计数器开始发挥它的神奇作用之前，需要将每个触发器清零，使得 $Q_4Q_3Q_2Q_1Q_0=00000$ ，这表示的是1个二进制数，也就是十进制的0，毫无疑问，所有触发器的输出端 $\bar{Q}$ 都为1  
如图9.8所示

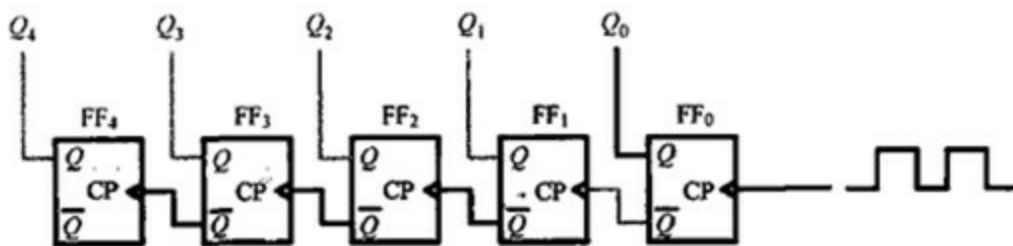


图9.8 当第一个脉冲到来后，计数器的值为00001

在第一个时钟脉冲到来的时候，在它的上升沿，最右边的那个触发器FF<sub>0</sub>翻转，使得它的输出 $Q=1$ ，于是 $Q_4Q_3Q_2Q_1Q_0=00001$ ，即十进制的1，

与此同时， $\bar{Q}$ 也从以前的1翻转为0，但由于它属于下降沿，因此不会对FF<sub>1</sub>及其他任何触发器造成影响

同时，当第二个脉冲到来时，FF<sub>0</sub>的 $Q$ 从1变成0，即 $Q_0=0$ ，而 $\bar{Q}$ 则从0变成1，对于FF<sub>1</sub>来说，FF<sub>0</sub>的 $\bar{Q}$ 从0变成1就意味着它的CP看到了一个上升沿，所以它也紧跟着触发，使得 $Q_1=1$ 。同时，FF<sub>1</sub>的 $\bar{Q}$ 也从1翻转到0，但这是一个下降沿，所以其他触发器不会跟着变化。此时， $Q_4Q_3Q_2Q_1Q_0=00010$ ，即十进制的2，如图9.9所示

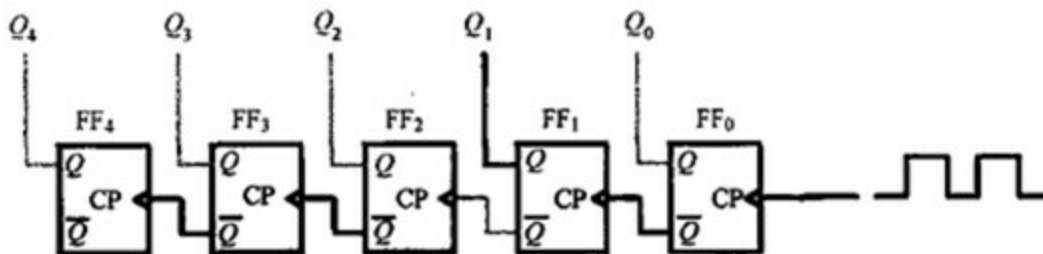


图9.9 当第二个脉冲到来后，计数器的值为00010

按相同的方法还可以分析，当第三个脉冲到来时， $Q_4Q_3Q_2Q_1Q_0=00011$ ，即十进制的3；第四个振荡器脉冲到来时， $Q_4Q_3Q_2Q_1Q_0=00100$ ，即十进制的4……就这样一直计数，直到

$Q_4Q_3Q_2Q_1Q_0=11111$ ，即十进制的31，如图9.10所示

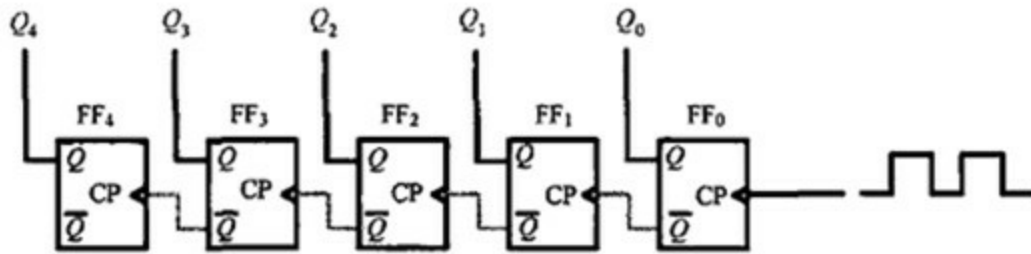


图9.10 当第31个脉冲到来后，计数器达到它所能表示的最大值11111

对于一个5比特的二进制数来说，11111，也就是十进制31已经是最大了，再大，用5个比特已经无法表示了。这意味着，如果想让计数器能统计更多的脉冲，唯一的办法就是增加这种反复触发器的数量。这样，如果用8个触发器组成这样的计时器，它可以累计255个脉冲（二进制数11111111）；如果用16个触发器的话，它可以累计65535个脉冲（二进制数1111111111111111），使用的触发器越多，能够累计的脉冲数就越大。

设想一下，如果计数器已经计数到最大，此时又来了一个脉冲，会怎样呢？这可以作为一个思考题，请读者结合图9.10自行分析。

计数器有广泛的用途，从广场上的倒计时牌到电子表，甚至每一台计算机的内部，都有它的用武之地，如果没有计数器，现代的计算机将无法自行工作，从下一章开始，我们的工作将证明这一点

计数器有广泛的用途，从广场上的倒计时牌到电子表，甚至每一台计算机的内部，都有它的用武之地，如果没有计数器，现代的计算机将无法自行工作，从下一章开始，我们的工作将证明这一点

## 第10章 用机器做一连串的加法

前面我们造出了加法机，几样简单的东西 – 继电器、开关、灯泡还有电线，就能做算术题，真是了不起。

和现代的计算机相比，用这台机器算加法，对人和机器都是一种考验。一方面机器需要忙碌的运转才能算出结果；另一方面人也不能闲着，得在旁边侍候着，扳动开关，输入数据，然后等着结果出来。这还不算什么，最让人心存疑惑的是，用它来算加法清空不如我们用纸和笔来得快，充其量也就是一个玩具，发明这东西有用吗？

当然是没有用 – 如果我们一直停留在这里原地不动。好在现实的情况并非如此。如果说现在你正在使用的计算机是一个智力健全的成年人，那么这台加法器就是一个刚出生的婴儿。从加法器到现代的计算机，这中间还有很多路要走，在这一章，我们将继续向这个宏伟的目标再迈进一小步

## 10.1 把一大堆数加起来

我们在前面制作的、能做加法的加法机就像汽车的发动机一样，将被当成一个用于制造计算机的零部件，所以称之为“加法器”或“加法部件”可能更符合它的身份。

名称不是大问题，不管叫什么，掩盖不了它其实就是一大堆全加器的事实。用加法器来算数学题，需要用一排开关拼成一个被加数，比如10011（十进制数19）；用另一排开关拼成一个加数，比如00101（十进制数5），然后加法器就会自动算出结果11000（十进制数24），如图10.1所示

注意，不像以前，在这里我们把加法器画成了两边高中间凹的桶形，这很形象地表明它要接收两路输入，相加后形成一路输出。如果这两排开关是我们头上的两只眼睛，那么加法器就是汇总和加工它们的大脑。

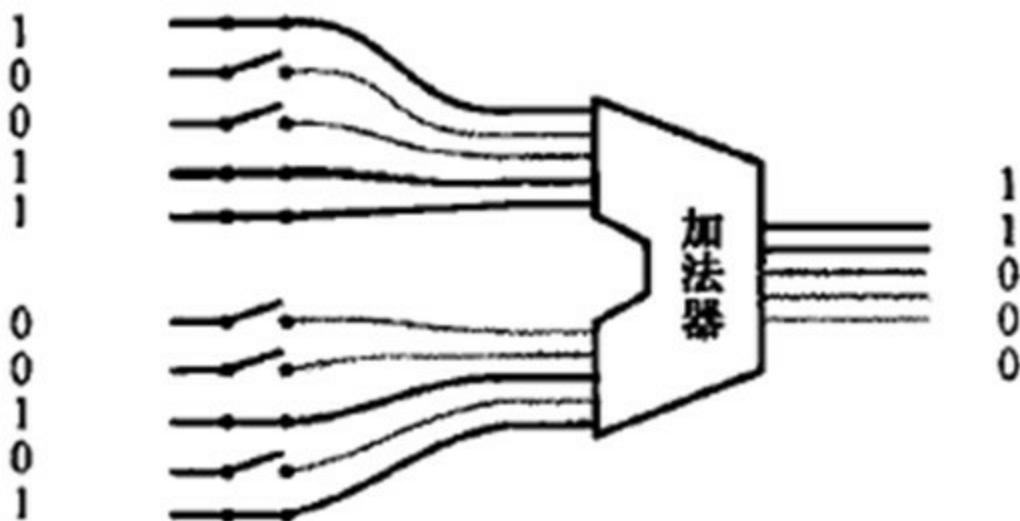


图10.1 使用两排开关分别给出被加数和加数，加法器就会算出结果

只做一次运算，仅仅把两个数相加，这很简单，怕的是有好几个数相加，比如：

$$10+5+7+2+6$$

首先，将10和5相加，也就是01010+00101，得出01111。01111只是一个中间结果，还要加上后面的数。所以还得再次折腾那两排开关，分别把它们扳成01111和00111，加法器算出结果，得到10110（十进制数22）

很显然，除了一开始，以后每次都有一个中间结果参与计算，都需要播弄一番开关，将上一次计算出来的结果再作为被加数或者加数输入到加法器中

这很不方便。即使连袖珍计算器也不用每次加两个数就要记住中间结果，再输入一次，相反你只需输入一个数，按一次+，再输入下一个数再按一次+.....就这样输入所有要相加的数即可得到结果，相比之下，我们的这个加法器在连续做加法时就显得很笨拙了。

这里面的奥秘是什么呢？

其实，即使是袖珍计算器也不能在不使用中间结果的情况下计算数学题。我们前面讲过了移位寄存器，同时也说明了所谓的“寄存器”是什么意思。在袖珍计算器内部，也有一个寄存器，可以暂时保存一个二进制数，中间结果就保存在这里。然后，每当你输入另一个数时，它就被安排来参与下一次计算，算出来的结果还放在这里。

保存一个二进制数？这里需要一个新的发明。一个触发器只能保存1个比特，但一个完整的二进制数通常包含好几个比特，是一个比特串。而且，二进制数中的所有比特都必须一直处理。

取决于想要保存的二进制数有多大，寄存器通常由好多个边沿D触发器共同组成。举个例子，如图10.2所示，这个寄存器包含了5个上升沿D触发器，所以能用来保存一个5比特长的二进制数

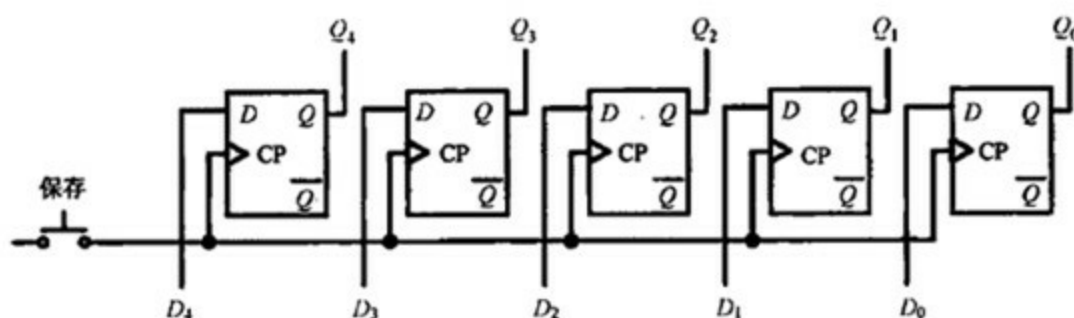


图10.2 使用多个触发器可以构成一个寄存器

不管一个二进制数包含多少个比特，要保存它，只需要把它的每一个比特都保存起来即可。所以，在这个例子中，被保存的二进制数，它的每一位分别进入 $D_0D_1D_2D_3D_4$ 这5根线；同时，所有触发器的CP端都连在一起，这样就可以接收同一个控制命令。一旦“保存”开关按下，在CP脉冲的上升沿，所有触发器同时开始干活，二进制数的每一位都在同一时间被保存起来，并立即出现在 $Q_0...Q_4$ 上

同样是为了方便，寄存器需要一个简明的图示。寄存器的符号如图10.3所示，注意那个三角形，这表明该寄存器只在CP脉冲的上升沿才会工作

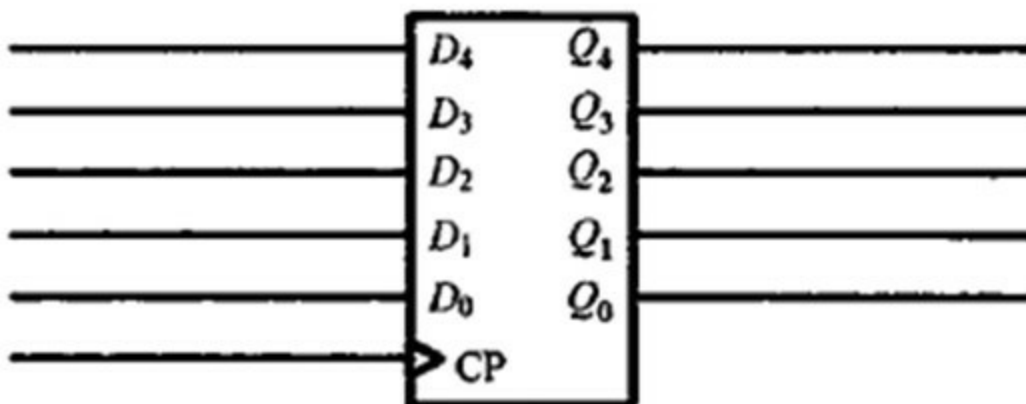


图10.3 寄存器的符号

寄存器登场后，做数学题时再也不用关心那些中间结果了。不过应该把它连接到加法器的什么位置呢？

如图10.4所示

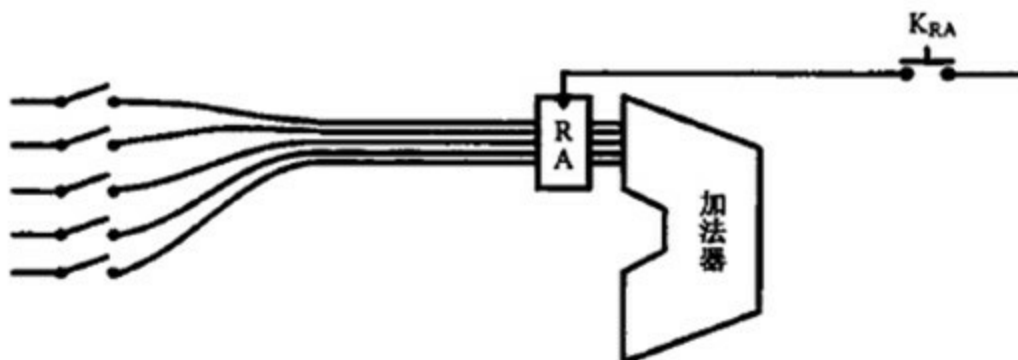


图10.4 数字先到达寄存器，再提供给加法器

RA就是我们刚制造的寄存器，它在加法器前排。按键开关 $K_{RA}$ 和RA的CP端相连，当我们用左边那一排开关扳出一个数后，如果按一下 $K_{RA}$ ，这个数就被锁住，与此同时，它把自己记住的内容输送到加法器并一起保持，作为第一个要相加的数。

现在，我们想用同一排开关向加法器提供另一个数，之所以不像以前那样分别用两排开关来提供被加数和加数，真正的原因到了后面我不说你能领悟。总之，用一排开关也很不错，如图10.5所示

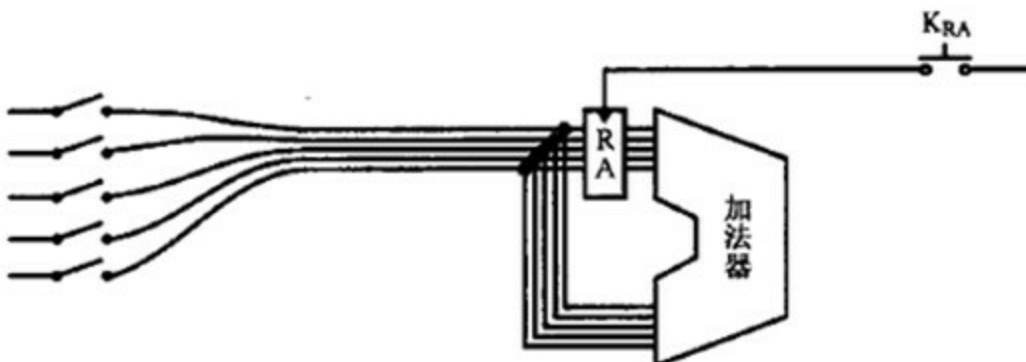


图10.5 用同一排开关既提供被加数，也提供加数



经过改造后，通过左边那排开关送进来的数可以到达寄存器RA，同时也被送到加法器的另一个输入端，取决于你的动机，如果你想把它保存到寄存器RA中，这按一下 $K_{RA}$ ；如果想用它的RA中的数相加，就什么也不做，结果自然会从加法器的输出端呈现。

当用开关摆出第二个数的同时，相加的结果也出来了（甚至在摆弄每个开关的同时，加法器也在进行计算）

要在往常，这个中间结果应该记下来，然后重新用开关输入运算器，和下一个数相加，但由于这样太麻烦，所以这个中间结果可以保存到寄存器RA中，这样就很自然地地为下一次计算准备好了一个中间结果

为了达到这个目的，一个可能的方案如图10.6所示，把加法器的输出同寄存器RA的输入端直接相连

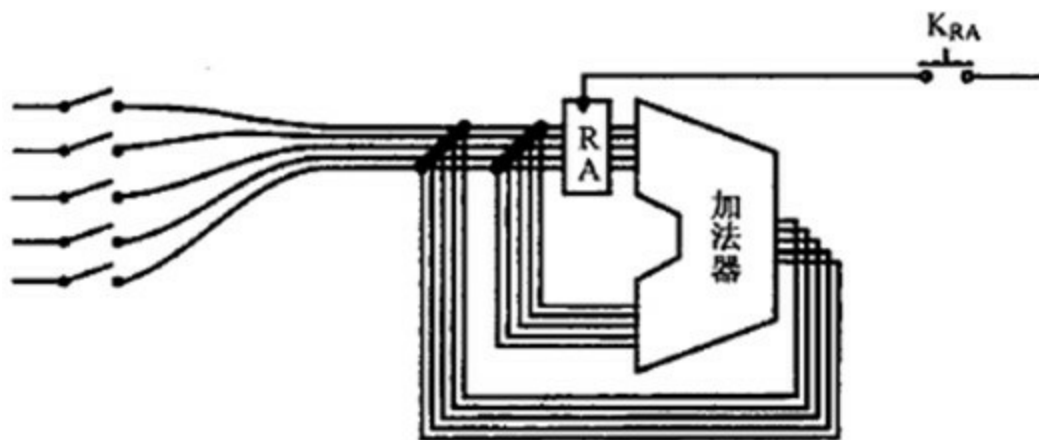


图10.6 加法器的计算结果应当返回寄存器中

想法不错，不过这里有几个麻烦，首先，左边那排开关和加法器的输出是直接相连的，都要走寄存器RA门前那段路。在逻辑电路里，大家共用的公共线路称为总线。想一想如果好几列火车都企图在同一时间通过一段铁轨时会发生什么，要是不把它们隔开，左边那排开关和加法器的输出都会抢着与寄存器RA说话。

其次从图中可以看出，即使不考虑电路冲突，计算结果在从加法器出来之后，不但会送往寄存器RA，还会再次进入自己的输入端，这理所当然地会形成一个反馈，而且是一边反馈，一边还在做加法，一切全乱套了。在这种情况下，还能指望保存在寄存器RA中的数是正确的吗？

要彻底解决这个问题，就必须重新设计整个电路，一个最简单的解决方案是使用电子开关，更多的时候，也称之为传输门。

## 10.2 轮流使用总线

最简单、最好理解的电子开关就是一只我们再熟悉不过的继电器，通过控制继电器线圈中电流的有无，可以间接地控制另一电路的通断。

二进制数的所有比特都是同时传输的，所以需要好几个继电器来分别接通或切断它的每一位。如图10.7所示，我们把所有继电器的线圈都接在同一个开关上，使它们同时吸合或释放，就可以达到目的

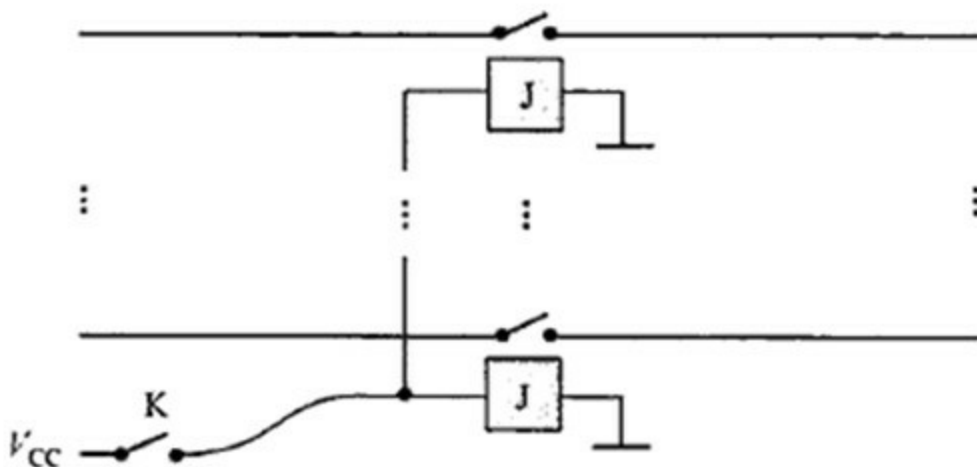


图10.7 用一只开关来控制多条线路的通断

采用继电器来制造这个隔离装置，当然直观又容易理解，但它并不是最好的材料，特别是在半导体技术出现后，使用半导体材料，我们可以制作更小、更便宜、更省电的这种电子开关来。对了，“电子开关”并不是一个专业的称谓，它真正的名字是“传输门”，实际上，它确实像门，打开它，信号可以从一边传到另一边，关上它，信号就不能传送了

原则上，要通过总线传送数据的任何一方都应该使用传输门以免互相干扰。如图10.8所示，左边那排开关通过传输门GA接入总线；加法器的输出则通过另一个传输门GB接入总线

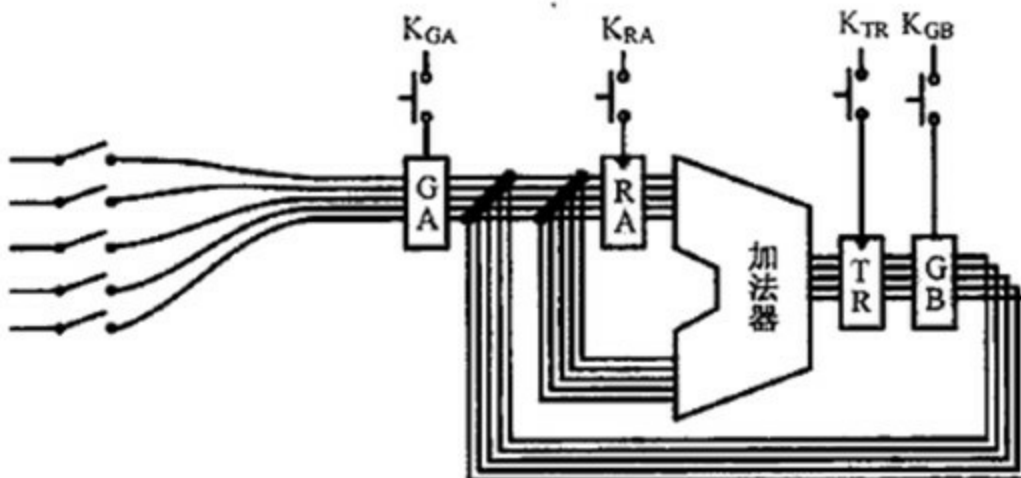


图10.8 完整的加法运算电路，它可以将多个数字相加

除此之外，可能已经发现该电路还多了一样东西。对了，确实多加了一样东西，这就是临时寄存器TR，它的RA是一模一样的，不同之处在于，它要用来临时保存加法器的计算结果。

用这个新设计的电路做加法是很有趣（当然，可能有些烦琐）。来看看它是如何工作的，在开始之前， $K_{GA}$ 、 $K_{GB}$ 、 $K_{RA}$ 、 $K_{TR}$ 都应当是断开的（强调这一点有些多余，因为它们都是按键开关，只要你不碰它们，它们就一直处于断开状态）

在前面，我们采用的例子是计算

$$10+5+7+2+6$$

为此，我们首先要做的是左边那排开关扳出第一个数10，并将其保存到寄存器RA中，这是开场动作，不妨称之为“装载”

装载的过程是这样的：假设数已经扳好了，接下来，按住 $K_{GA}$ 不要松开，使传输门GA打开，于是数据到达寄存器RA；接着，再按一下 $K_{RA}$ 将数据锁进RA，最后松开 $K_{GA}$

当然了，从图10.8可以看出，这个数不但到达寄存器RA，还到达加法器的另一个输入端，以及传输门GB。同时，加法器也一直在工作，但因为传输门GB没有打开，这里不会出乱子

这是个单一的过程，但却需要两只手操作，同时还有一个手法问题。什么手法呢？那就是不允许同时按下 $K_{GA}$ 和 $K_{RA}$ 。当按下 $K_{GA}$ 时，数据还没有稳定下来，要是RA在这时工作，它保存的数据就很有可能是错的

装载过程结束了，第一个数10已经位于寄存器RA中了，现在，我们要用第二个数与它相加。这需要再次扳动那排开关，得到第二个数5，然后，按住 $K_{GA}$ 不要松开，使5进入加法器另一个输入端。加法器是自动即时相加的，它会立即计算出相加的结果，此时，按一下 $K_{TR}$ 将其

保存到临时寄存器TR中，然后松开 $K_{GA}$

因为要做一连串的计算，所以当前的计算结果还必须参与下一次计算，这意味着要把数据从临时寄存器TR移动到RA

通常情况下，传输门GA是断开的，所以不用担心数据冲突，直接按住 $K_{GB}$ 不要松手，使计算结果从寄存器TR通过GB流向RA；接着，按一下 $K_{RA}$ 将数据锁存，最后将 $K_{GB}$ 松开。

有趣的是，一旦数据从传输门GB流出来，它不但会等待RA将其保存，同时也流向加法器的另一个输入端，并和RA中原有的数相加，不过不用担心，寄存器TR会将结果拦住，以防止形成一个反馈

很明显，在这道加法题中，除了第一个数字10需要预先保存到寄存器RA之外，从第二个数字5开始，一直到最后一个数字6，所有数字在相加时的操作过程都是一样的，都要经历开关扳数，相加并保存到寄存器TR，然后从TR移动到RA的过程，这个过程可简单地称谓“相加”，当最后一个数加完之后，最终的结果仍然在寄存器RA中

当然，寄存器只有5位，能表示的最大数是11111（十进制数31），如果要加的数很多，而且每个数都很大的话，它可能会产生一个进位，但这个进位将被加法器丢掉，这将在RA中得到不正确的结果。但在这道题里，数字又小又少，就是保证它正常运行即可。我们的目标是先让它能工作起来，再想办法完善它。

## 10.3 简化操作过程

用我们发明的计算器做一连串加法，比我们原先所想的要麻烦多了。不过话说回来，这也很有趣

有趣归有趣，还是希望操作越简单越好。要不然，一手一只开关，左右开弓，真别扭，而且不注意还会搞错，本来是要按这个，却按了那个。所以需要进行技术革新

注意，我们的真实想法是去掉那些操作开关（左边那排用来输入数据的开关还得留着），用别的方法来完成这些操作

开关的作用是接通或断开电源，使电路从0变成1，或从1变成0。开关可以做到的事，逻辑电路也能做到。如图10.9所示

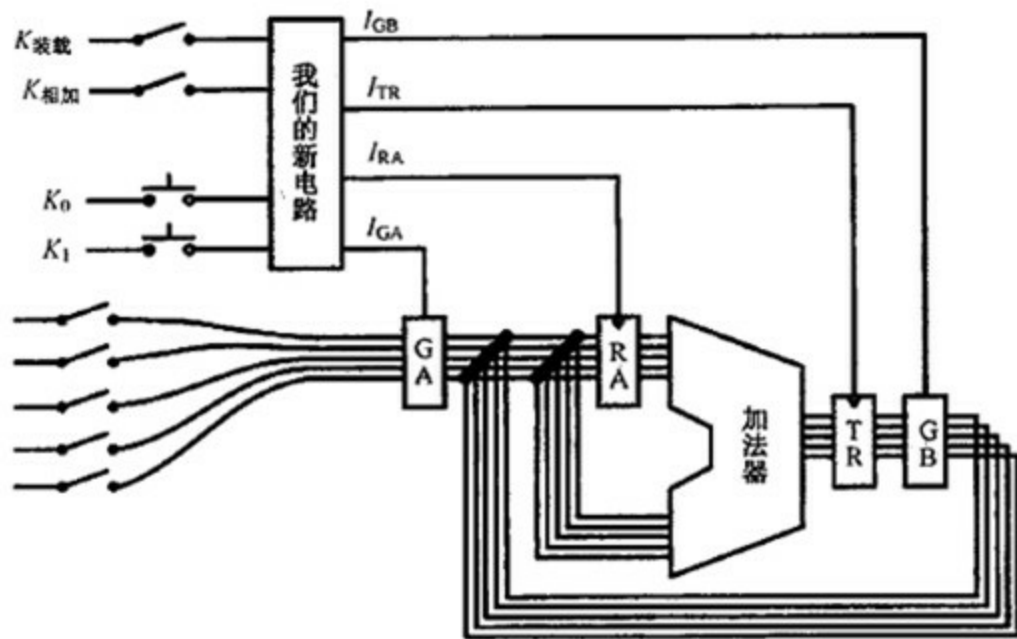


图10.9 一个使相加操作过程变得有规律的电路设计

我们重新发明了一个逻辑电路，这是个新生事物，一时之间不知道叫什么了，姑且叫“我们的新电路”吧。该电路有4个输出，用于代替以前的那些开关

一个普通的逻辑电路而已，它不可能古怪精灵到能自动按正确的时间和顺序产生输出。现在我们所能做的，就是给它提供相应的输入

设计逻辑电路是门学问，要是你掌握了它，就会知道，要得到相同的输出，可以有很多不同的设计方法。在这里，我们用4个开关作为输入，表面上看，我们只是给原来的开关换了位置，拆东墙补西墙，换汤不换药。不要误会，这只是表面现象，很快就会发现它的妙处。

由于刚刚做过加法，我们知道，要把一大堆数加起来，首先要执

行“装载”动作，后面都是“相加”，这两件事，虽然有关联，但毕竟是两码事，所以我们用两个开关 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 来指明要做哪件事。合上 $K_{\text{装载}}$ ，断开 $K_{\text{相加}}$ 就是要装载一个数到寄存器RA中；断开 $K_{\text{装载}}$ ，合上 $K_{\text{相加}}$ 就表明要开始做加法了

还有一点，注意这两个开关的类型，它们都是闸刀开关，断开和闭合都需要分别手动一次，采用这种类型的开关，是因为它们在后面的操作中基本上不需要扳来扳去

一旦 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 被扳到合适的状态，就意味着政策方针已定，剩下的事情就是分步骤来完成这个目标。逻辑电路不是精灵，不改变它的输入，它就不会有相应的输出，而 $K_0$ 和 $K_1$ 的作用就在于此。当按顺序分别按下 $K_0$ 和 $K_1$ 时，就可以完成“装载”或“相加”的全过程。我们会喜欢这种操作方式的，因为它有规律可循，只要按顺序分别按动两个开关即可

注意与 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 不同， $K_0$ 和 $K_1$ 是按键开关，之所以采用这种开关是因为它们需要频繁操作。

不过为了完成“装载”或“相加”，只用两个开关 $K_0$ 和 $K_1$ 够吗？答案是刚好。为了证明这一点，同时也为了设计这个逻辑电路，我们需要详细地定义一下它的工作状态，设计一张表格，并从中得到逻辑表达式

首先不管 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 的状态如何，只要 $K_0$ 和 $K_1$ 中的任何一个没有接通，所有的输出都必须为0，如图10.1所示，这可以禁止错误的操作

表10.1  $K_0$ 和 $K_1$ 未按下时的状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{GA}$	$I_{RA}$	$I_{TR}$	$I_{GB}$
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0

不过，同样是 $K_{\text{装载}}$ 闭合、 $K_{\text{相加}}$ 断开的情况下（这表明我们要开始装载），如果按下 $K_0$ 则在逻辑电路的输出端，只有 $I_{GA}$ 和 $I_{RA}$ 为1，传输门GA打开，寄存器RA执行锁存动作，保存从左边那排开关上来的数据，如表10.2所示

表10.2 执行装载功能，当 $K_0$ 按下时的状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{GA}$	$I_{RA}$	$I_{TR}$	$I_{GB}$
1	0	1	0	1	1	0	0

这里有个问题，寄存器RA必须等待从GA来的数据稳定下来才能动作。换句话说， $I_{RA}$ 应该比 $I_{GA}$ 晚一点为1才行，在足球场上，守门员得判断球从哪个角度过来才能采取动作，如果别人刚准备射门，守门员就开始动作，这就太心急了

那为什么不换一种方法，按下 $K_0$ 的时候， $I_{GA}$ 先为1；按下 $K_1$ 的时

候， $I_{RA}$  再为1呢？

原因很容易理解，逻辑电路的输入直接对应着输出，要想在两个不同的输入之间保持某个输出的连贯性，就既困难，也不合理。

这确实是个问题，但眼下没办法解决，只能先放一放

可以看出，要做“装载”的工作，只需要按一下 $K_0$ 就行， $K_1$ 是多余的，它存在的原因是因为后面的相加过程需要两个步骤，必须使用两个开关，既然是这样，我们强迫译码电路在 $K_1$ 按下时什么也不做，如表10.3所示

表10.3 执行装载功能，当 $K_1$ 按下时的状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{GA}$	$I_{RA}$	$I_{TR}$	$I_{GB}$
1	0	0	1	0	0	0	0

装载过程完成，接下来是“相加”了，也就是将另一个数和寄存器RA中的数相加，这需要闭合 $K_{\text{相加}}$ ，断开 $K_{\text{装载}}$ 。此时，如果按一下 $K_0$ ，则 $I_{GA}=I_{TR}=1$ ，数据通过GA进入加法器，与寄存器RA中的数相加之后锁进临时寄存器IR，如表10.4所示：

表10.4 执行相加功能，当 $K_0$ 按下时的状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{GA}$	$I_{RA}$	$I_{TR}$	$I_{GB}$
0	1	1	0	1	0	1	0

这里同样有脉冲先后的问题，还是先不管它

一个完整的相加过程还包括把结果返回寄存器RA的动作，这需要接着按一下 $K_1$ ，使 $I_{GB}=I_{RA}=1$ ，数据离开临时寄存器TR，穿过传输门GB，到达寄存器RA后被锁存，如表10.5所示

表10.5 执行相加功能，当 $K_1$ 按下时的状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{GA}$	$I_{RA}$	$I_{TR}$	$I_{GB}$
0	1	0	1	0	1	0	1

这又是一个脉冲先后的问题，我们马上就会把它解决掉

在这个电路上工作，重要的是必须按先后顺序按下 $K_0$ 和 $K_1$ ，一个按下的时候，另一个已经自动弹开，这意味着，无论 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 的状态如何，只要是 $K_0$ 和 $K_1$ 同时按下的情况，逻辑电路的输出一律为0，如表10.6所示

表10.6 当 $K_0$ 和 $K_1$ 都按下时的状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{GA}$	$I_{RA}$	$I_{TR}$	$I_{GB}$
0	0	1	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0

不过，即使 $K_0$ 和 $K_1$ 只有一个为1，如果 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 都断开，或者都闭合，这也不是我们所希望的，所以一律将电路的所有输出都置0，如表

## 10.7所示

表10.7 其他一些无效的开关状态

$K_{\text{装载}}$	$K_{\text{相加}}$	$K_0$	$K_1$	$I_{\text{GA}}$	$I_{\text{RA}}$	$I_{\text{TR}}$	$I_{\text{GB}}$
	0	0	0	1	0	0	0
	0	0	1	0	0	0	0
	1	1	0	1	0	0	0
	1	1	1	0	0	0	0

4个开关，每个开关都有0和1两种状态。这样，它们就有16种组合，已经分别列在前面7个表中。如何从这张大的真值表中得到 $I_{\text{GA}}$ 、 $I_{\text{RA}}$ 、 $I_{\text{TR}}$ 、 $I_{\text{GB}}$ 的逻辑表达式，并根据这些逻辑表达式来组装这个“我们的新电路”应该不是问题了吧。



## 10.4 这就是传说中的控制器

理想中，在这台机器上做加法是很有趣的，因为它不但简化了操作，而且操作过程中还具有很强的规律性。还是前面那个数学题

$$10+5+7+2+6$$

回到图10.9，要计算这些数字的总和，首先合上 $K_{\text{装载}}$ ，断开 $K_{\text{相加}}$ ，用左边那排开关扳出数字10，然后分别按一下 $K_0$ 和 $K_1$ 。这里，10就被保存到寄存器RA中了

接着，断开 $K_{\text{装载}}$ ，合上 $K_{\text{相加}}$ ，用左边那排开关扳出数字5，再分别按一下 $K_0$ 和 $K_1$ ，于是就得到了相加的结果15，它位于寄存器RA中

不要再理会 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 了，让它们维持现状，毕竟后面全是加法，现在，再用左边那排开关扳出第三个数7，再次分别按下 $K_0$ 和 $K_1$ ，于是又会在寄存器RA中得到本次相加的结果22

按顺序，后面要加的数是2和6，但不管后面还有多少数字要加，操作过程是一样的。

理想毕竟是理想，正如前面指出的那样，这个电路实际上是有缺陷的。比如，它总是让 $I_{\text{GB}}$ 和 $I_{\text{RA}}$ 同时为1，以便把相加的结果从寄存器TR移动（复制）到RA，但后者前者晚一点为1才能保证可靠性。

要让“我们的新电路”正常工作，不可避免地要继续完善它。有趣的是，改造后的电路不但工作起来更可靠，同时还能获得一个额外的好处，那就是它现在只用三个开关，比以前少一个，最重要的是，操作起来更简单

$K_0$ 和 $K_1$ 的作用不过是产生脉冲，在我们的操作下，先是 $K_0=0$ ，然后 $K_1=1$ ，接着又是 $K_0=1$ .....这很容易让我们想起循环移位寄存器。既然移位寄存器也能达到相同的效果，那就没有理由不在我们的电路中使用。

为了取代 $K_0$ 和 $K_1$ ，需要用两个上升沿D触发器来制造一个循环移位寄存器，并把它加入我们的电路中，如图10.10所示

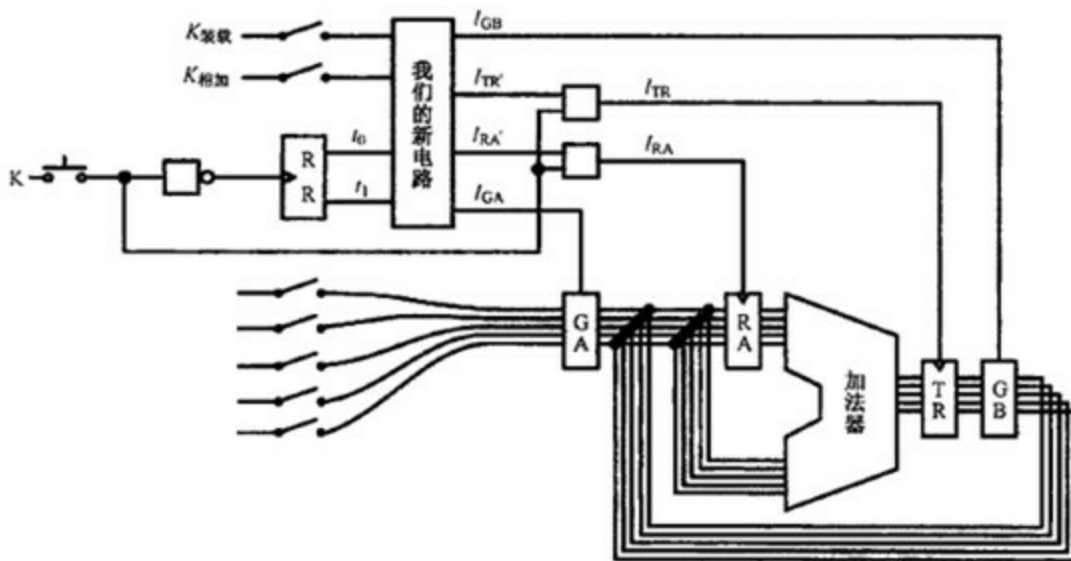


图10.10 使用循环移位寄存器来简化装载和相加过程

图中，RR就是我们刚刚加进来的循环移位寄存器，它有两个输出 $t_0$ 和 $t_1$ ，取代了原先的开关 $K_0$ 和 $K_1$

另一个比较大的改动是电路的输出部分，在这个新方案中， $I_{RA}$ 和 $I_{TR}$ ，我们重新定义了 $I_{RA}$ 和 $I_{TR}$ ，现在，它们分别来自两个与门的输出

现在来分析一下这个新设计是如何运作的。开始之前，应当先设置循环移位寄存器RR的初始状态，使它的两个输出 $t_0$ 和 $t_1$ 分别为1和0

现在要开始计算一连串的加法了，首先是装载第一个数到寄存器RA中，这需要用左边那排开关来扳出这个数字

接着闭合 $K_{\text{装载}}$  断开 $K_{\text{相加}}$ ，此时，由于RR的输出 $t_0=1$ ，按以前的设计， $I_{GA}=I_{RA'}=1$ ，传输门GA打开，如图10.11所示

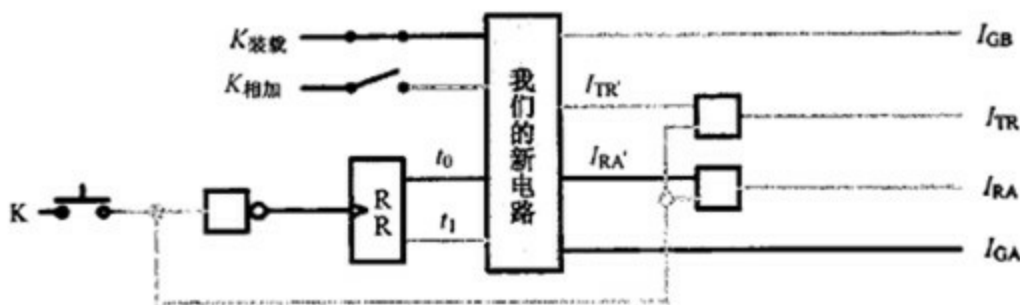


图10.11 装载数据的过程 (1)

现在数据已经送到寄存器RA嘴边了，但它带吃不了，因为K还没有按下，所以通过与门输出的 $I_{RA}=0$ ，这是有意的，因为寄存器RA需要等数据稳定后才能运作

现在，按下开关K，如图10.12所示

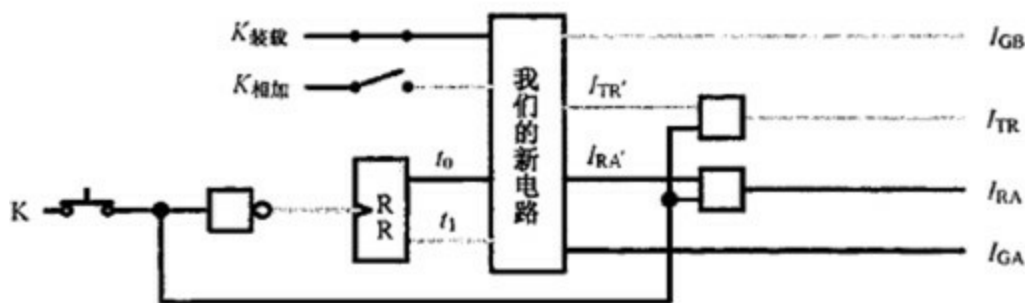


图10.12 装载数据的过程 (2)

随着K的接通，将同时产生两路脉冲，第一路通过非门到达循环移位寄存器RR，遗憾的是这是一个由高到低的下降沿，RR不会理睬，这也意味着，“我们的新电路”仍然保持原来的输出不变

与此同时，另一路脉冲被直接送到与 $I_{RA}$ 相连的与门，使得 $I_{RA}$ 从0变为1，在它的上升沿，寄存器RA将数据锁存

很显然，这么复杂只是希望在K闭合和松开的过程中， $I_{RA}$ 比 $I_{GA}$ 慢半拍出现，好安全地将数据锁住。现在，隐患已经消除了

上面说的是当K按下时所发生的事情，看得出来，电路构造巧妙、而工作过程绝对精彩，不过我们不能一直按着K，毕竟还有很多事情要做

因为是按键开关，当一松手，K马上弹开，如图10.13所示

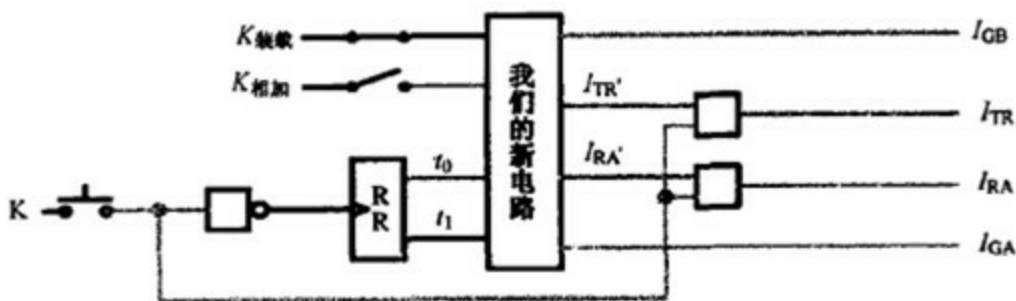


图10.13 装载数据的过程 (3)

按键松开的瞬间，它产生的0经非门后变为1，这个变化对循环移位寄存器RR来说是上升沿。寄存器RR循环移位一次 $t_0=0$ ， $t_1=1$ ，根据前面的设计，所有的输出都是0，自动进入下一个步骤。

尽管完成“装载”的工作只需按一次开关K就可以，但在前面设计这个电路时，我们预设的是做一件事需要按两次，已经按了一次，现在再按一次

如图10.14所示

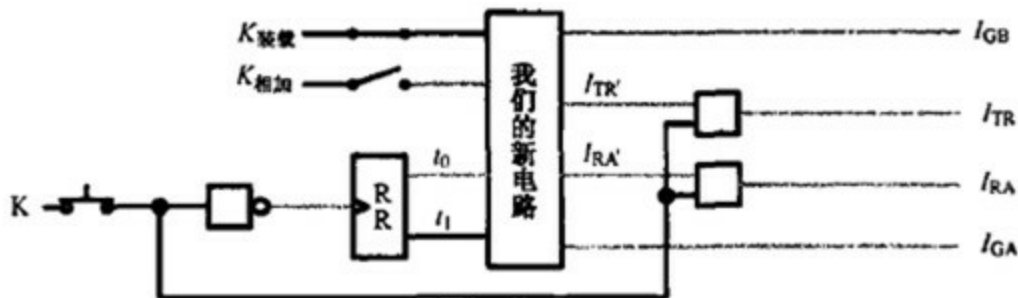


图10.14 装载数据的过程 (4)

因为所有的输出都是0，所以即使K被按下，它们也不可能突然就变成1，同时K按下时，对RR来说是下降沿，这不会对它有任何触动

但和前面一样，当K松开时，RR又循环移位一次，使得 $t_0=1$ ， $t_1=0$ ，这又回到了一开始，也就是图10.11所示的状态，这意味着可以再来一次“装载”的过程

现在有两个选择：一，要是觉得刚才装载的数据不对，想重新装载一次，可以直接再按两次K；二，如果准备开始做加法，进行“相加”，就断开 $K_{\text{装载}}$ 合上 $K_{\text{相加}}$ 。

如果是“相加”，断开 $K_{\text{装载}}$ 合上 $K_{\text{相加}}$ ，并用那排开关准备好加数。如图10.15所示，真奇怪这个电路马上改变了状态，根据前面的设计， $I_{GA}=I_{TR}=1$ ，传输门GA打开，提前使另一个数进入加法器并开始计算

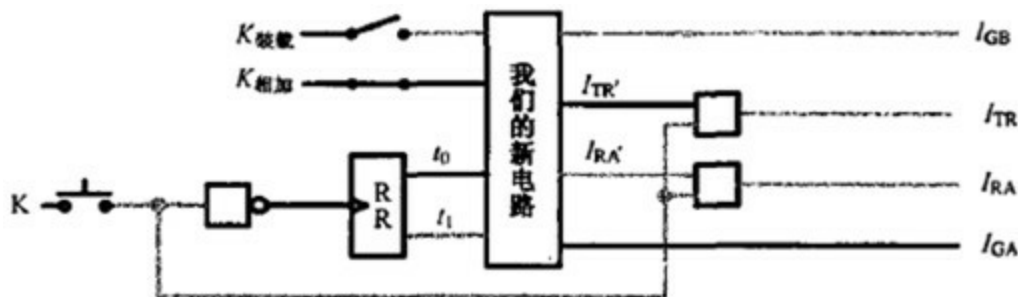


图10.15 相加过程 (1)

现在按一下K执行第一个运作，和前面一样，先是 $I_{TR}$ 和来自K的上升沿脉冲一起，使得 $I_{TR}$ 从0翻转到1，临时寄存器TR锁存相加的结果，如图10.16所示

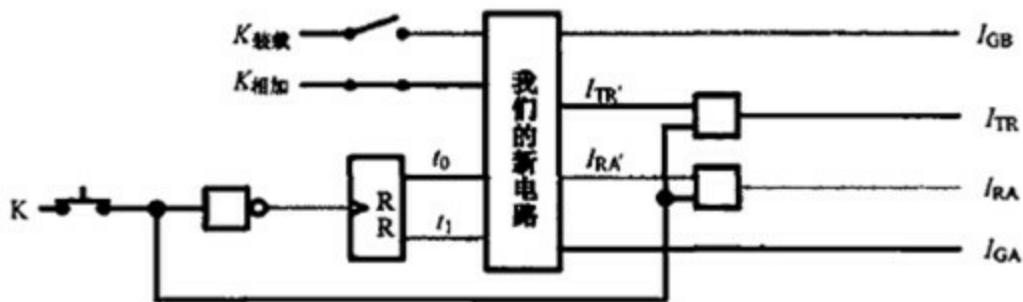


图10.16 相加过程 (2)

完整的“相加”过程需要两步，这是第一步，已经得到了相加的结

果，现在需要把这个结果移动（复制）到寄存器RA中。因为电路就是这样设计的，所以当K松开时，RR再循环移位一次， $t_0=0$ ， $t_1=1$ 于是 $I_{GB}=I_{RA'}=1$ ，为下一次K按下时将计算结果锁存在寄存器RA中提前做准备，如图10.17所示

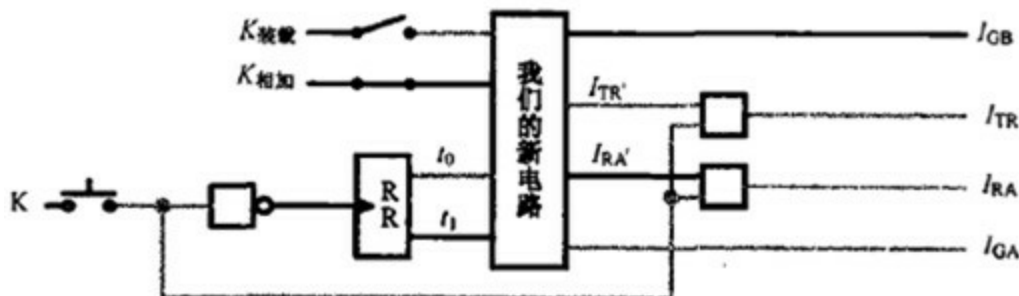


图10.17 相加过程（3）

现在数据已经穿过传输门GB，安全地来到寄存器RA，只要再按一下K，使得 $I_{RA'}$ 和从K来的上升沿脉冲一起，把 $I_{RA}$ 从0翻转到1即可，在这个上升沿，寄存器RA锁存结果，如图10.18所示

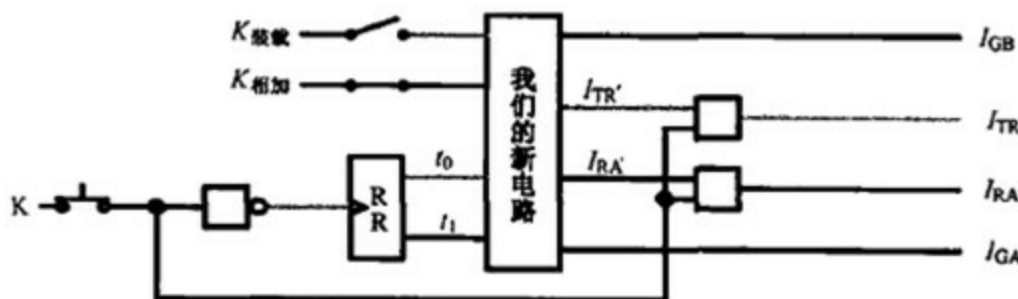


图10.18 相加过程（4）

松开K时，循环移位寄存器从非门那里得到一个从0到1的上升沿脉冲，所以再次循环移位，使得 $t_0=1$ ， $t_1=0$ ，于是整个电路又回到“相加”过程的最开始，也就是前面的图10.15

这意味着什么？因为要做一连串加法，除了最开始要将第一个数装载到寄存器RA之外，其他都是单纯的相加，所以这意味着，从现在开始，可以按照“用开关扳数 - 按两次K - 用开关扳数 - 按两次K - .....”这样的模式将所有的数加完，最终的结果就在寄存器RA中

计算机中有个控制器，使计算机按规定的步骤计算。事实上，我们的这个电路就是一个控制器，当然它太简单了简陋了，还需要进一步完善，不过这是以后的工作了，我们的当务之急是解决另一件同样很麻烦的事情

# 第11章 全自动加法计算机

能够将一大堆数加起来，这不错，麻烦的是要来回拨弄一大堆开关，而且要想用好它，还要掌握二进制记数法。

我们的目标是从头制造一台现代的计算机，在不需要人工干预的情况下自动计算。当然，要达到这个目标还有一段距离。现在的问题是，要把一大堆数加起来，靠手工操作不是很方便，而且说不定什么时候不注意把开关扳错了，还得从头再来一遍。要是有成百上千个数相加，而这个错误恰巧发生在只要完成最后一个数就可以得出结果的时候，多么悲惨

为了避免这样的惨剧，能不能把所有的数提前存起来，然后让机器自己一个一个取出数进行相加呢？尤其是考虑到这样做还有一个特别的好处，那就是如果有些数错了，可以单独修改，然后让机器再从头计算一遍，反正它是自动的。能不能发明出这样一种计算机呢？

通常一个能保存很多二进制数的东西叫存储器

## 11.1 咸鸭蛋坛子和存储器

提起存储器，马上会想到袋子、盒子、坛子、罐子、箱子、柜子……，实际上这些容器的确是存储器。但我们需要的是能保存二进制数的存储器

尽管保存的内容不一样，但所有的存储器都有一个特点，那就是它们通常都只有一个口，通过这个口，可以把东西放进去或把里面的东西取出来。基于同样的原因，一个能保存许多二进制数的存储器也应当只有一个出入口，可以把它想象成一个箱子，里面装了许多二进制数，但这个箱子只有一个口，一次只能放入一个或取出一个。

最小存储容量的存储器就是我们熟悉的上升沿D触发器，它可以而且仅仅只能保存1个比特。

存储器最好是用来保存一大堆二进制数，而不是一大堆单个的比特，就像书柜应该用来存放大量的书而不是堆满了没有装订在一起的纸。但存储单个比特的触发器是制造大容量存储器的基础，要是有能力保存一个比特，同样可以保存更多的比特。

考虑到存储器和咸鸭蛋坛子的相似性，它应该只有一个出入口，但存储器比咸鸭蛋坛子的制作烦琐多了，因为存储器必须使用传输门，如图11.1所示，这个存储器由一个上升沿D触发器构成，只能保存一个比特

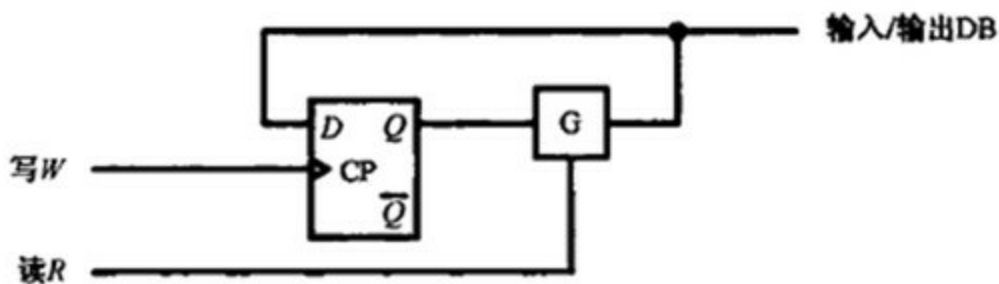


图11.1 具有唯一输入/输出线的存储器，它可以保存或读出一个比特

对于D触发器来说，G是传输门，作用是将D触发器的输出Q同外部接通或断开。

和往常不一样，对于存储器来说，工程师们习惯将保存一个数称为“写”（Write），而从存储器里取出一个数则称为“读”（Read），不管是“读”还是“写”，一律称为“访问”（Access）。所以，W和R分别用于这个存储器中写入或读出一个比特。

平时，W和R都为0，这个存储器什么也不做，既不能写入，也读不出比特，因为传输门G是断开的，而触发器CP端也没有接到任何有效的指示。

这是存储器的默认工作状态。如果在DB上准备好一个比特，然后使W从平时的0变为1（上升沿）。如图11.2所示

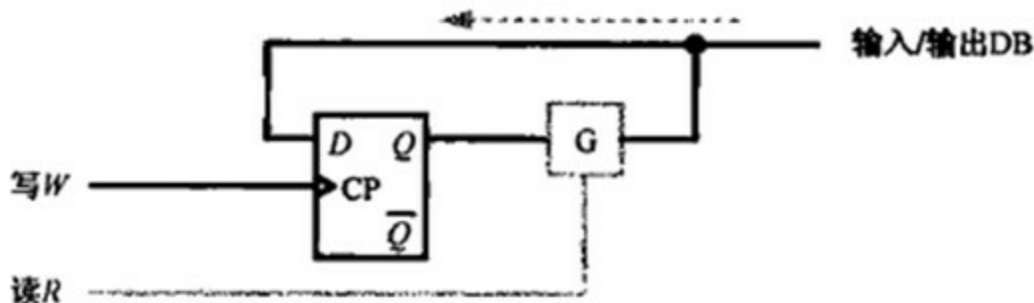


图11.2 写入一个比特的原理

在W脉冲上升沿，比特被D触发器保存。之后，应当使W重新为0

写入比特时，应使R保持它平时为0的状态，毕竟是“写”而不是“读”，在这种情况下，传输门G是断开的，来自DB的输入和Q上的输出不会遇到一起。否则，一定会发生冲突。这种场面不是我们想要的。

相反，如果是要读出数据，那么如图11.3所示，必须使R=1以打开传输门，触发器Q端的比特被送入DB总线。

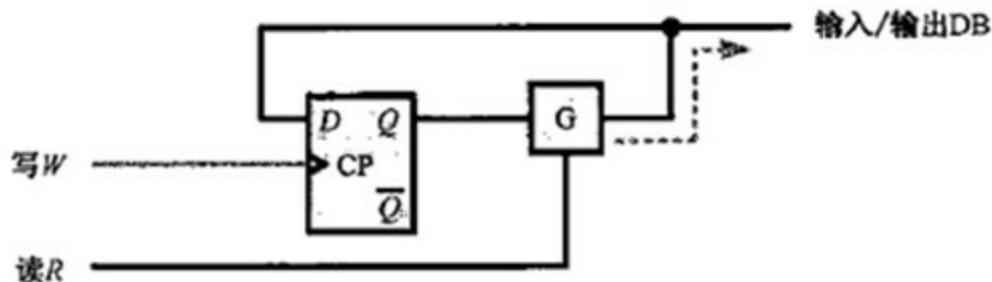


图11.3 读出一个比特的原理

没错！这个输出Q同时也被送入触发器自己的输入端D。但没关系，因为在读的过程中，W必须为0。最后，如果确信外部已经将这个比特取走，必须使R=0

我们已经暗示过，W和R不能同时从0变为1，因为这很奇怪、很无理的要求，意思是我既想读又想写。要是非这样做，那么存储器会做一个很奇怪的运作：吞食自己的输出，这没有任何意义。

只保存一个比特，这个存储器的容量未免太小了。但这个存储器可以做为一个基本单元，制造更大容量的存储器。所以，我们姑且称它为比特单元，为了方便后面的讲解，我们给它一个示意图，如图11.4所示



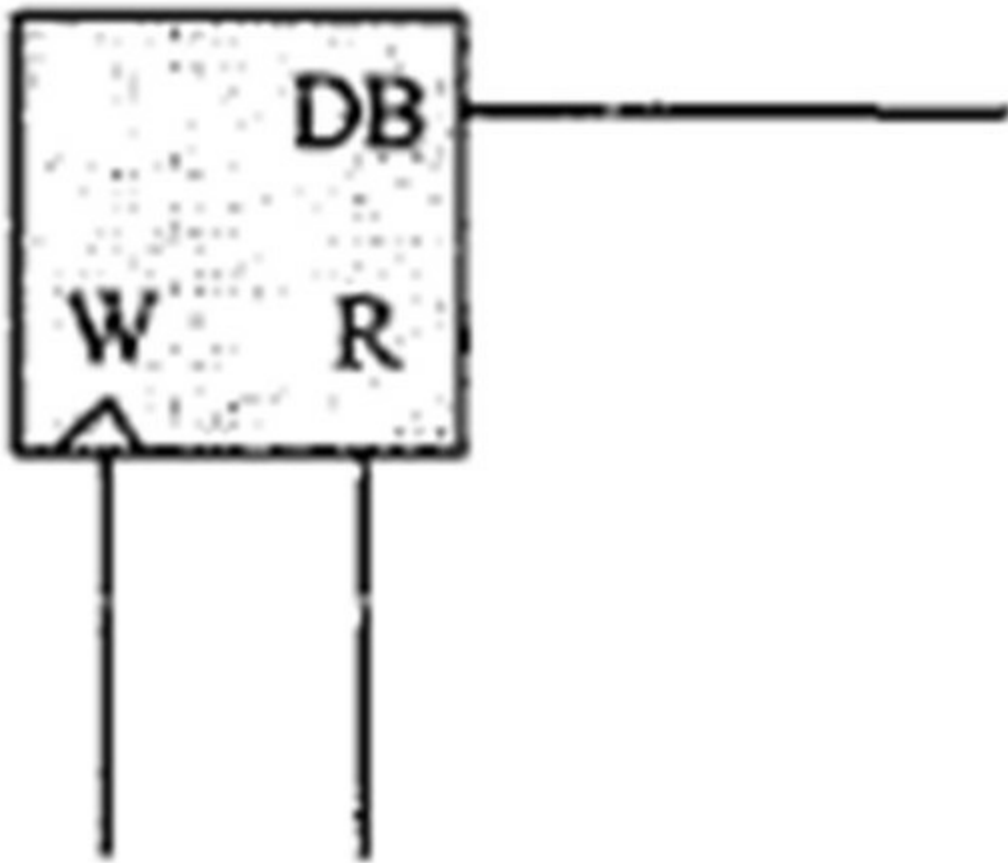


图11.4 比特单元示意图

毕竟是和电路有关的东西，再加上刚刚学过它的工作原理，只读/写1个比特几乎没有什么用处，我们制造存储器，是希望它能保存完整的二进制数。

这个愿望不难实现，如图11.5所示，因为一个二进制数通常包含许多比特，是一个比特串，所以可以把很多比特单元并排组织起来，以容纳该二进制数的每一位

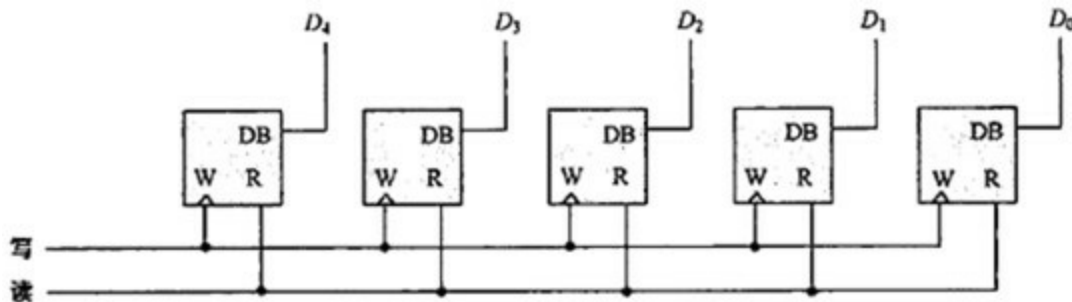


图11.5 可以读/写单个5位二进制数的存储器

使用多少个这样的比特单元，取决于你要保存的二进制数有多大，换句话说，它包含了多少个比特，在这个盒子中，我们使用了5个比特单元，所以它只能保存5位的二进制数

这样，当要写一个二进制数时，可以把它的每一位分别放在 $D_0$ - $D_4$ 这5根线上，保持“讯”线为0不变，并通过“写”线发出一个上升沿脉冲，这时，它们将分别被独立地保存起来；相反，当要读出这个二进制数时，只需要使“写”线保持0不变，“读”线为1即可，它的每一位会自动出现在 $D_0$ - $D_4$ 上

直观上来说，这5个比特单元并排在一起，很像一层楼。既然一层楼可以保存一个二进制数，那么，多盖几层，就可以保存好几个二进制数了，不是吗？如图11.6所示，一共盖了4层，可以保存4个二进制数。当然，要想保存更多，完全可以盖更多层。

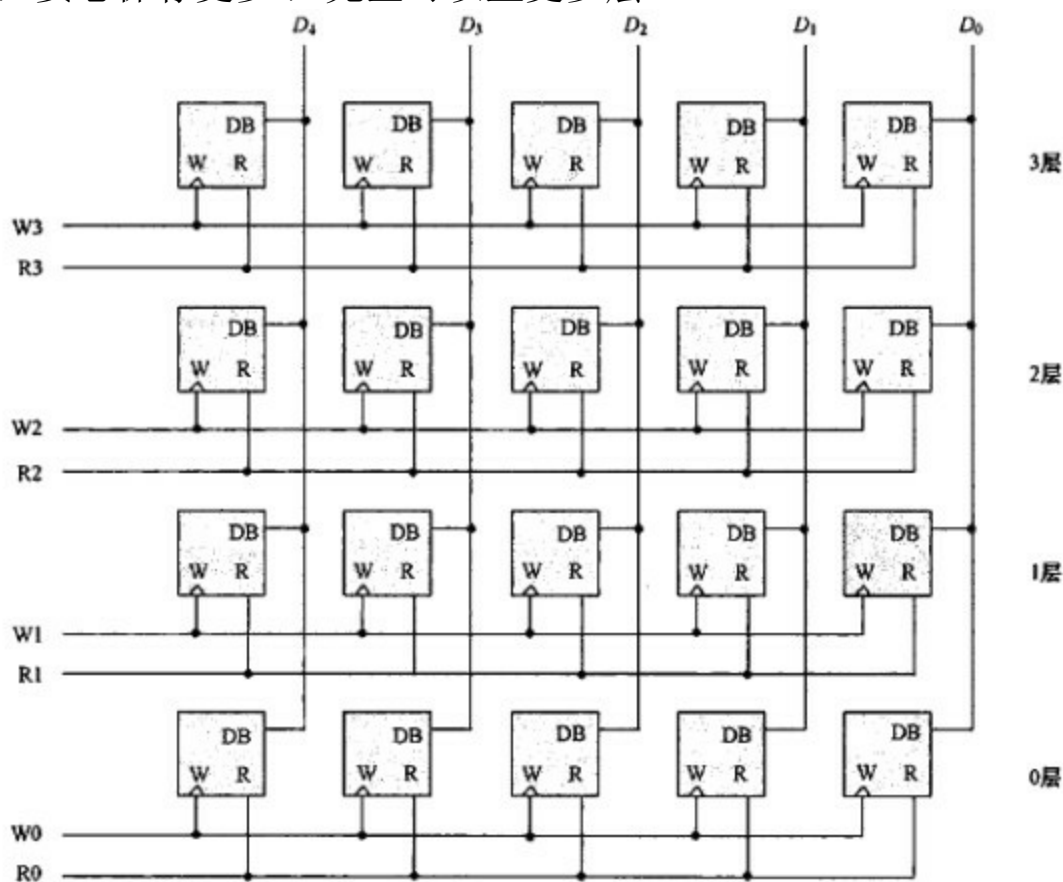


图11.6 存储器示例，它可以保存4个二进制数，每个数最多包含5个比特

既然把它看成楼层，那么，为了指明把一个二进制数保存到哪一层，或者从第几层读出来，需要给每一层编号，不过和平时不同，我们不是从一楼开始编号，而是从地下室开始，所以这4个楼层按顺序应该是0，1，2，3层。

每个楼层都有自己的读线和写线。比如，0层的读线是 $R_0$ ，写线是 $W_0$ ；第1层的读线是 $R_1$ ，写线是 $W_1$ ，其他楼层依次类推。平时，每一层的读线和写线都为0，既写不进也读不出。要想向这个存储器里写入一个二进制数，只需要使相应楼层的写线0翻转为1即可，读的时候与此

类似。当任何一个楼层正在读/写时，其他楼层都处于休眠状态，既不能读，也不能写。所以它们不会互相干扰

为了操纵读线和写线，给它们通电，或者使它们断电，可以使用开关，但问题是少量的楼层还好办，要是有成千上万层，就要使用成千上万个开关、成千上万条电线.....

除了体积和数量上的因素外，想想看，要是准备向99999层写入一个二进制数，还要先找到第99999号开关W99999.....难道不能把开关省略了吗？

每个楼层不是都有编号吗？我们可以通过指定编号的方法来告诉存储器，我们要访问的是哪个楼层

通过编号来指定存储单元需要一排开关，通过扳动开关来拼成二进制数，每个二进制数表示楼层的编号。这是非常巧妙的方法，假如我们有15个开关，就可以拼出任何一个15位的二进制数，从000000000000000到111111111111111，也就是十进制的0到32767，换句话说，只需要15个开关就可以指定32767个楼层中的任何一个，不用为每个楼层都配备两个开关了

为了举例还是看前面那个例子吧，它只有4个楼层，编号为0，1，2，3，分别对应二进制数的00，01，10，11，都是2比特的二进制，正好需要两个开关就足够了

要想用两个开关来从4个楼层中选出一个，实际上该如何做呢？要如何设计电路连接呢？当然，答案是发明一个新逻辑电路，用来选择一个楼层

如图11.7所示

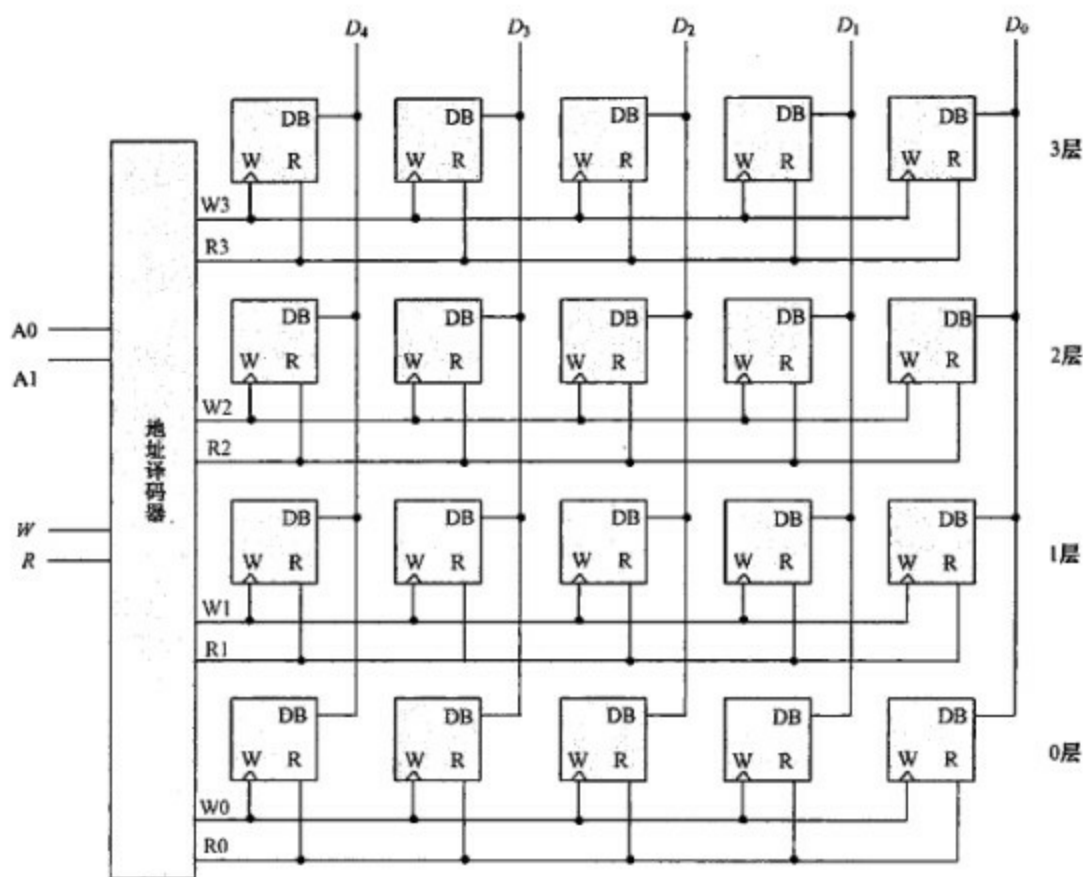


图11.7 地址译码器对于简化存储器设计是必不可少的

这个新的逻辑电路叫做“地址译码器”。名字的含义稍后再说，重要的是  $A_0$  和  $A_1$  共同用来指定一个楼层，如果  $A_1A_0=00$ ，表示选择的是第0层；如果  $A_1A_0=11$ ，表示选择的是第3层

该逻辑电路还有另外两个输入，也就是  $R$  和  $W$ ，它们用于指明对所选择的楼层进行何种操作 – 读还是写。要是  $R$  和  $W$  都为0，那么，不管选择了哪个楼层都没用，因为此时  $R_0, W_0, R_1, W_1, R_2, W_2, R_3, W_3$  都为0

但假如选择了第2个楼层，即  $A_1A_0=10$ ，那么，当  $W=1$  时， $W_2=1$ ；相反，当  $R=1$  时， $R_2=1$ ，这就是该逻辑电路的工作原理。

对于存储器来说，不管是楼层也好，编号也罢，这都是打比方。要想进入计算机行业，必须改口要说“地址”。这实际是借用了生活中的词汇，好处是容易理解和接受，通过指定一个地址，就可以从这个存储器的某个位置取出一个数字或写入一个数字

地址仅仅是一个编号、一个门牌号码，它指向存储器内部的一个小空间，这是真正用于保存数据的地方，这个地方就叫存储单元。每个地址对应一个存储单元。

一旦通过  $A_0$  和  $A_1$  给出一个存储单元的地址，那么结合  $R$  和  $W$  的输入

情况，就能得到另外一种上形式的输出，使相应的存储单元开始读或写，这实际上就是一个转换或翻译的过程，这就是为什么称之为“地址译码器”的原因

地址译码器需要详细定义一张符合其工作状态的真值表，并依据该真值表写出各项输出的逻辑表达式，在这本书里已经做过多次了，请自己试一试

这个复杂的存储器要是把它包装起来放进盒子里，用起来就方便多了。

如图11.8所示，这是封装之后的存储器，它有4个地址引线A0-A3，可以访问0000-1111这16个存储单元，这就是它的存储容量，另外，它有D<sub>0</sub>-D<sub>4</sub>五根数据线，这意味着它每次可写入或读出一个5比特的二进制数

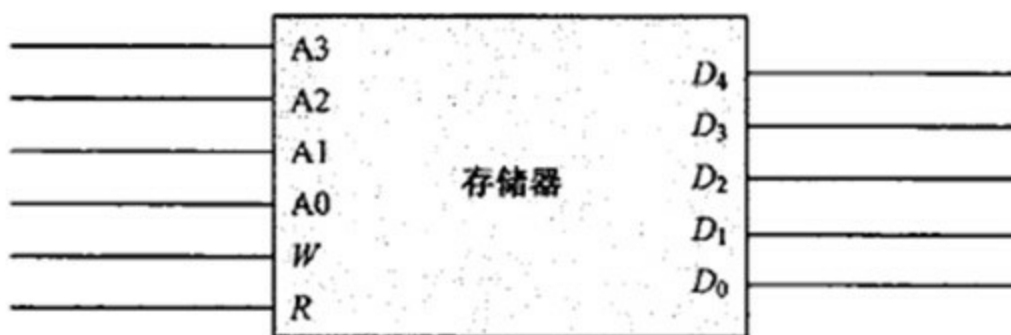


图11.8 封装后的存储器整体外观

在生活中，多数存储器，例如咸鸭蛋坛子，都只能按顺序一点点地存放或取出，上面的蛋不拿走，下面的就拿不出来。但我们的这个存储器，可以随机地、任意地决定要访问哪个存储单元，不管访问哪个存储单元，所花的时间都一样，和地址没有关系。正是因为这样，通常称谓“随机访问存储器”，或“自由存取存储器”，用英文来说就是Random Access Memory，简称RAM。由于组成它的细胞是触发器，而这种东西怕断电，不管它记下的是什么，只要一断电就完蛋了，因此属于易失性存储器

## 11.2 磁芯存储器

用触发器来制造存储器，这似乎是很轻松、很容易的事情，而且从理论上来说无论多大的存储器，都可以用触发器堆叠出来。要是存储容量很小，采用触发器来制造还是可以理解的，当存储容量变得很大时，这种做法就不行了

一般的，保存1个比特的成本是好几个电子管或晶体管，假如需要5个（实际这根本不够），那么，按一个二进制类有8比特来算，要保存100个二进制数至少需要 $5 \times 8 \times 100 = 4000$ 个，还没有把地址译码器算在内。

老实说一个存储器只能保存100个二进制数，现在根本没人要，但在电子管和晶体管的时代，却能让科学家在梦里笑醒。想想，好几千个晶体管，连线有多复杂，体积有多大，需要消耗多少电。当然还包括研究经费。20世纪30年代末，也就是电子管发明30年后，一只性能良好的电子管还要10美元才能买到。而那个时候，正是计算机开始跑步前进、需要大容量存储器来为它提供原料的时候。

困难很多，而存储器也是必要的。没办法人们绞尽脑汁，想了很多古怪的方法来制造存储器。在这些曾经用过的东西当中，占统治地位的是一种叫做磁芯的东西。

我们都知道电磁学，也明白电和磁的关系，对于你见钢这样的东西，容易被磁化，即使本身没有磁性，但在和磁石接触之后会变成一块磁铁，这叫做剩磁。和钢一样，磁芯用铁氧体材料制成，像一个圆环，外径通常在0.2-2mm之间，比一粒芝麻大不了多少

用磁芯来保存数据是利用了电流能产生磁场，有些东西在磁化之后会产生剩磁，同时，磁场反过来也可以产生电流的原理。而且更重要的是，不同的电流方向将使磁芯按不同的方向磁化，换句话说，电流从左流向右，和从右流向左，这两种情况下磁芯的南北极是截然相反的，这也意味着，如果把一种剩磁状态看成0，则另一种状态就是1，那么磁芯可以用来保存1个二进制比特。

如图11.9所示，在磁芯中穿一根电线，叫做驱动线，用来向磁芯中写入一个比特，所以称这根电线为写入线也是可以的，但最好还是称为驱动线，因为读出比特的时候也要用到它，要达到写入一个比特的目的，需要控制电流的方向。

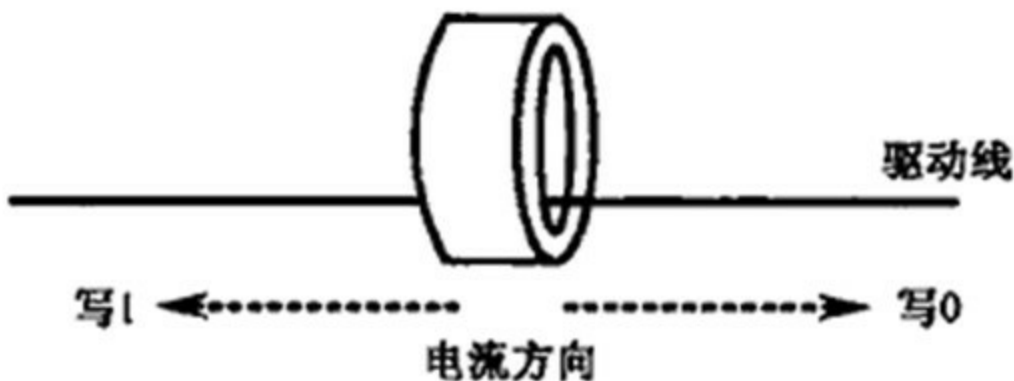


图11.9 电流方向决定了磁芯的磁化方向

向磁芯中写入一个比特是比较简单明了的，但要把这个比特读出来就很古怪了。需要在它里面穿另一根线，叫做读出线，如图11.10所示

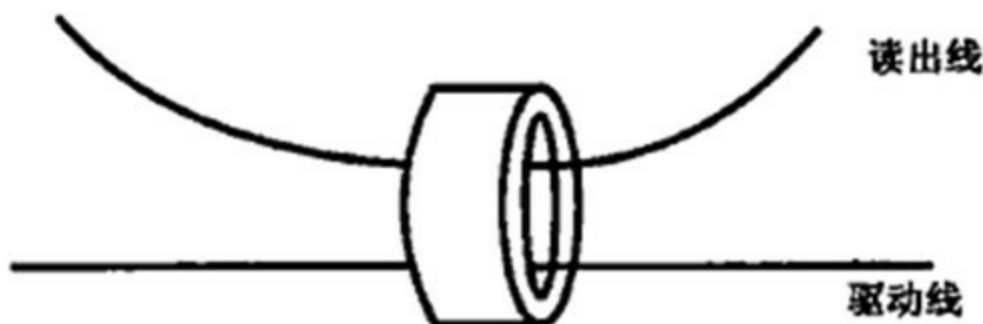


图11.10 磁芯的工作原理

现在你面临着和法拉第一样的困惑：只是在磁场中穿一根电线，这毫无用处，静止的导体不会产生电流，如何在读出线中感应出这个比特数据呢？

读出比特的过程要麻烦一些，但很有趣，方法是用驱动线向磁芯写入一个比特0，如果磁芯中保存的本来就是0，那么这个写入电流不会对原有的磁场产生太大的影响，以至于读出线上感应出的电压很小，这表示读出的是0；但如果磁芯中原来保存的是1，那么这个写入电流将使磁芯的磁场翻转，这种变化非常强烈，这个大幅度变化的磁场将在读出线上感应出较高的电压，这表示读出的是1

可以看到，磁芯的读出是破坏性的，如果读出的是1，那么读取之后磁芯的状态将会变成0，这需要重新将读出的1写回磁芯，这是磁芯存储器比较麻烦的地方

从磁芯中读出的电压很小，而且不是我们需要的方波，好在有电子管和晶体管，可以将其放大进行规整，使之符合要求。和触发器不同，磁芯有一个明显的好处，就是断电后，也能维持写入的数据。

磁芯在全世界应用了好几十年，被实践证明是那个时代最好的存储器材，发明它的人是美国华裔科学家王安博士。

1949年10月21日，王安申请了磁芯存储器的专利，几年之后，又创

办了自己的公司，生产小型商用计算机和文字处理机，事业兴旺，盛极一时。1988年，王安被列入美国发明家名人堂，自1901年创建以来，只有爱迪生和68人入选发明家名人堂。非常遗憾的是，从20世纪80年代后期开始，王安的公司迅速衰落，最后销声匿迹。

中国的第一台电子计算机用的也是磁芯存储器



# 11.3 先存储，后计算

但愿你还没有忘了为什么我们要发明存储器，现在让我们在第10章的基础上继续发明创造，来搞清楚如何让机器自动取数，然后计算。

因为我们要计算

$$10+5+7+2+6$$

而且，我们还想预先把这些要加的数都写入存储器，然后再一个一个取出相加，为此，我们可以使用图11.8那样的存储器。一方面，这5个数都不大，每个数的长度都不超过5个比特；另一方面，要加起来的数只有5个，而这个存储器却有16个单元，空间绰绰有余。

在开始做加法之前，先要把上面那5个要加起来的数写到这个存储器里，如果没有特殊的原因，所有的二进制数都应该从存储器的顶端，也就是地址0000开始一个挨一个存放，如图11.11所示

地址	存储内容	十进制
0000	01010	10
0001	00101	5
0010	00111	7
0011	00010	2
0100	00110	6
0101	未用	
⋮	⋮	
1111	未用	

图11.11 将所有要加起来的数顺序存放在连续的存储单元里

为此，可能要用4个开关来形成地址，再用5个开关拼成写入的二进制数，最后一个开关连在存储器的W端，用来向存储器下达写入命令。从地址0000开始，每当一个地址和一个二进制数准备好后，按一下W开关，就这样操作，直到把所有的数都按顺序写入存储器，在这个过程中

难免会出点小差错，但是存储器的好处就是你可以反复修改任意一个地址里的内容

因为我们只是把5个数相加，所以前5个地址，从0000到0100，里面的内容是我们特意写入的，其他地址，也就是从0101到1111，没有使用。

正常情况下，要把刚才写进去的数一个一个读出来，同样需要给出地址，然后使 $R=1$ ，不过我们想做的古怪些，希望能够用最省事的方法连续操作，把它们按顺序一个一个地取出来，毕竟目标是将它们按顺序相加，当然，最好是像第10章那样，拍拍开关就能做到。

其实不太难，图11.12就是我们给出的方案

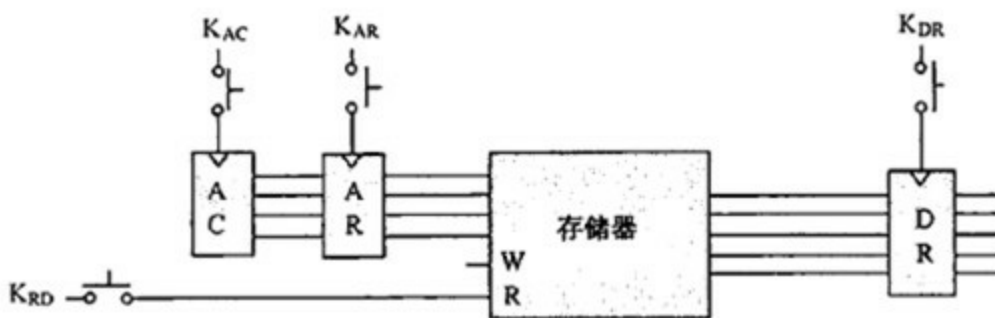


图11.12 顺序地从存储器里取数的电路方案

如图所示，假设存储器里已经存放了我们要加起来的5个二进制数；AC是计数器，用以提供访问存储器的地址，所以称为地址计数器。一开始它的内容是0000，每按一次 $K_{AC}$ ，它就在原来的基础上自动加一，以得到访问下一个存储单元所需要的地址；AR是一个寄存器，用来临时存放存储器地址，称为地址寄存器。看起来AR有些多余，似乎用AC给存储器提供地址更直接。但很快会发现，不把计数器AC直接和存储器相连，而是由AR负责转交是非常有道理的

很明显， $K_{RD}$ 的作用是给寄存器发出命令，要求它将数据送出。注意存储器的W端没有使用，因为我们现在只是要读，所以将它悬空，让它一直为0

在存储器的数据端，数据寄存器DR用于暂存读出的数据，除了名字不同之外，它和普通的寄存器没有什么两样。存储器只负责把客人送出门外，所以数据应当在DR中稍事休息，等待进一步的指示，以决定自己应该动身前往何处

介绍了这些组件后，现在可以接二连三地往外取数了，在开始之前，先想办法把地址计数器AC清零，以指向地址0000，然后执行以下操作：

(1) 按一下 $K_{AR}$ ，地址计数器AC当前的值0000被AR锁存，并提供给存储器

(2) 先按住 $K_{RD}$ ，不要松开，再按一下 $K_{DR}$ ，这时，数据送出，并被 $DR$ 保存；最后，松开 $K_{RD}$

(3) 按一下 $K_{AC}$ ，地址计数器加一以指向下一个地址，为再次从存储器里读数据做准备

到此，存储器里的第一个数就被取出来了。如果要接着取第二个数，第三个数.....，重复上面的3个操作步骤即可

在第10章，我们已经学会如何只用一个按键开关、一个循环移位寄存器和一个逻辑电路简单地操控，把一大堆数加起来。道理都是一样的，按这种思路，同样可以制作一个控制器，用它来简化取数过程，如图11.13所示

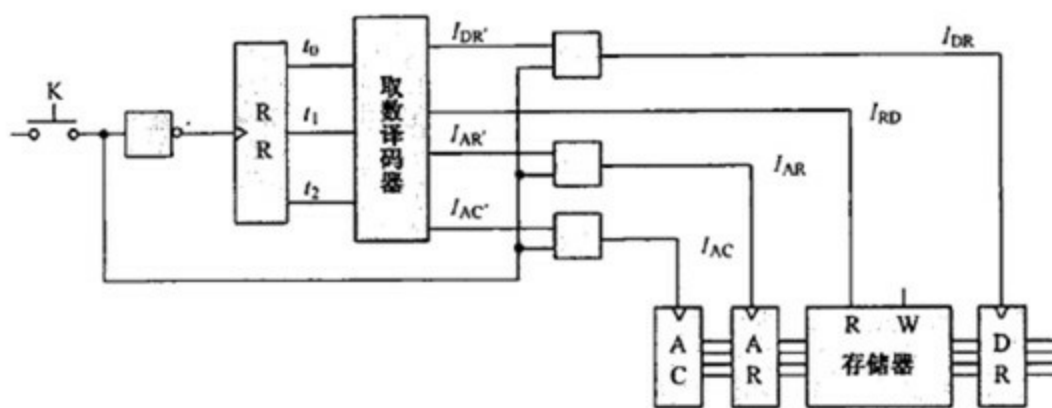


图11.13 用一只开关依次将数取出

图中， $RR$ 就是我们据说的循环移位寄存器，和第10章的循环移位寄存器不同，此处的循环移位寄存器有3个输出 $t_0$ ,  $t_1$ 和 $t_2$ ，毕竟刚才我们已经看到了，把一个数从存储器里取出来，需要经历三个不同的步骤

同样是在第10章，为了制造控制器，我们发明了一个逻辑电路——“我们的新电路”。现在看来，它其实就是一个译码器，把一种形式的输入转换翻译成另外一种形式的输出，在我们现在的这个控制电路里，同样需要一种类似的译码器，因为发明它的目的是从存储器里取数，所以称之为“取数译码器”

电路刚启动时， $t_0=1$ ， $t_1$ 和 $t_2$ 都为0，此时，只有 $I_{AR'}=1$ ，如图11.14所示

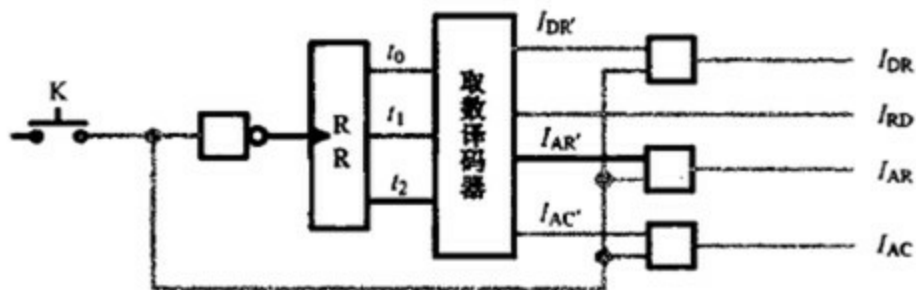


图11.14 取数控制器原理 (1)

一旦开关K被按下，则 $I_{AR'}$ 和从K来的1一起，通过与门使得 $I_{AR}$ 从0翻转到1，于是地址寄存器AR将地址计数器的地址保存，并提供给存储器。对于循环移位寄存器RR来说，从非门得到的是一个下降沿，它所能做的就是不予理睬，如图11.15所示

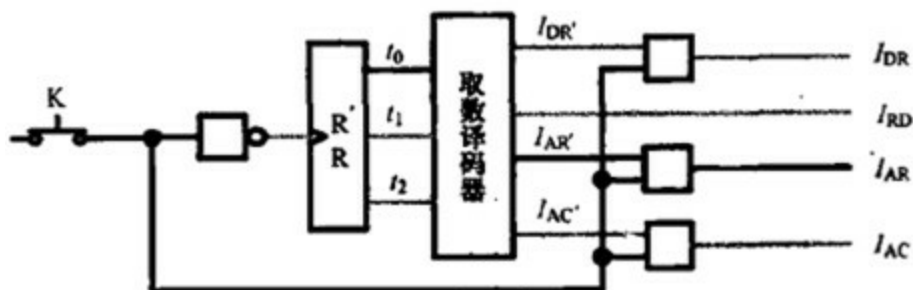


图11.15 取数控制器原理 (2)

当K松开时，RR得到一个上升沿脉冲，于是 $t_0=0$ ， $t_1=1$ ， $t_2=0$ ，并因此使得 $I_{RD}=I_{DR'}=1$ ，如图11.16所示。此时，存储器开始向外送出数据，但还不能被DR寄存器保存

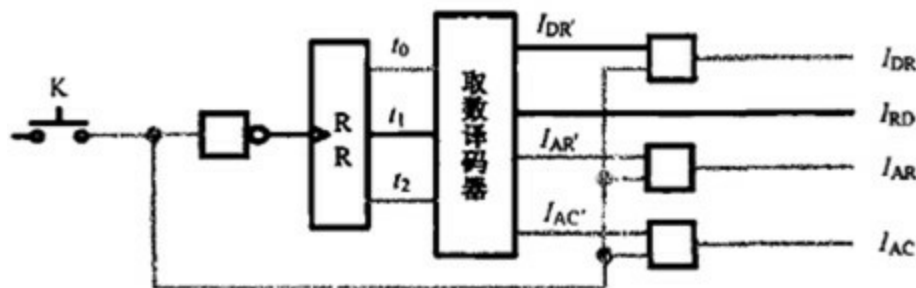


图11.16 取数控制器原理 (3)

当K第二次按下时，RR依然无动于衷，但 $I_{DR}$ 却立即从原来的0翻转为1，于是寄存器DR将存储器送出的数字保存起来，如图11.17所示

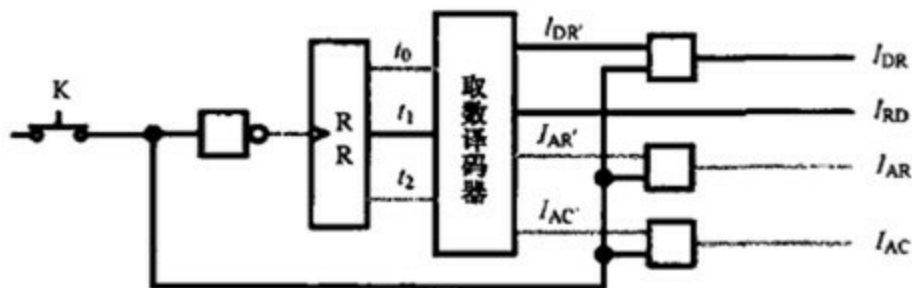


图11.17 取数控制器原理 (4)

K再次松开后， $t_0=t_1=0$ 而 $t_2=1$ ，于是 $I_{AC'}=1$ ，如图11.18所示

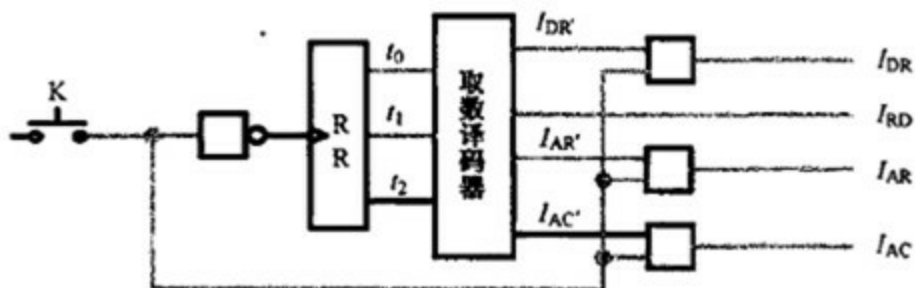


图11.18 取数控制器原理 (5)

现在我们第三次按下K，这将使得 $I_{AC}$ 从0翻转到1，于是地址计数器自动加一，以指向下一个存储单元的地址，如图11.19所示

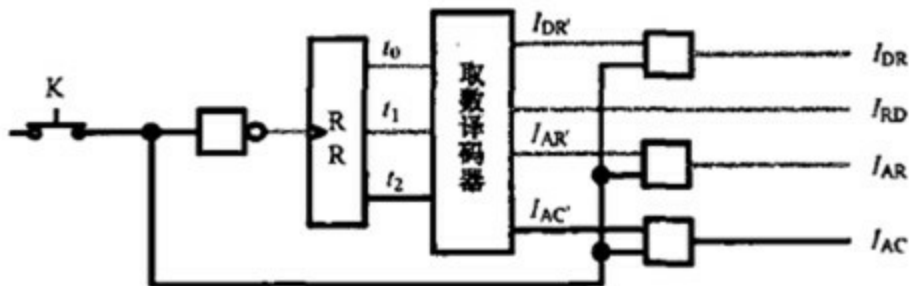


图11.19 取数控制器原理 (6)

一旦K第三次松开，RR将再次循环移位一次，使得 $t_0=1$ ，而 $t_1=t_2=0$ ，这以回到一开始，即图11.14，换句话说，如果再连续按三次K，将会把下一个存储单元里的数取出来

## 11.4 半自动操作

我们的目标是用机器计算一连串的加法。还记得为什么要发明存储器吗？那是因为我们有一个美好的愿望 – 把所有要相加的数提前保存起来，然后，机器自动把它们取出来相加。目标就在眼前，让我们继续前进

在第10章中，所有参与相加的数都是用开关得到的，现在，存储器和加法器可以合在一起，实现从存储器里不断取数，然后相加的功能。如图11.20所示

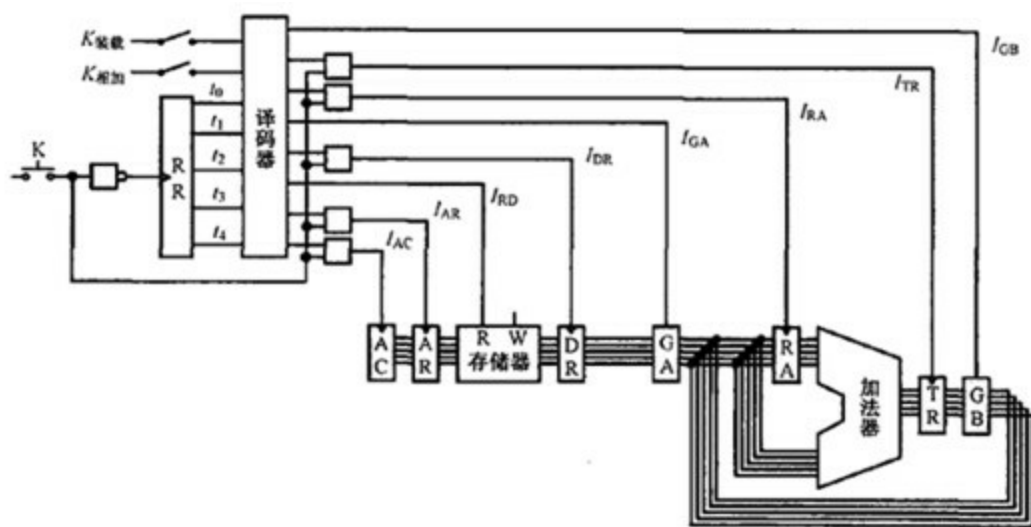


图11.20 部分实现自动化的连续加法电路

新的控制电路有些复杂，但它的工作原理很好理解，从存储器取出一个二进制数（并将地址计数器加一），需要三个步骤；而在第10章，把一个二进制数装载到寄存器RA中，或者用另一个数与RA中的数相加，分别需要两个步骤。现在，取数和做加法已经二合一了，这就是为什么循环移位寄存器RR现在有5个输出 $t_0$ ~ $t_4$ 的原因

两个功能的合并意味着需要重新设计一个新的逻辑电路来产生各种输出，这就是图中所示“译码器”，所幸的是，这对现在的你来说并不困难，是不是？

在这台新机器上操作，和以前并无二致，开关K依然是使用的主要道具，每次当1从 $t_0$ 循环移位到 $t_3$ 时（换句话说 $t_0$ - $t_3$ 阶段），将完成从存储器里取数到寄存器DR，并将地址自动加一的功能，这个阶段的功能永远是固定的，与 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 无关，不受它们的影响，这一点在设计译码器时需要注意

$t_3$ 和 $t_4$ 阶段, 执行的功能取决于 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 的状态, 如果 $K_{\text{装载}}=1$ 而 $K_{\text{相加}}$

=0，则将DR中的数装载到寄存器RA中；如果 $K_{\text{装载}}=0$ 而 $K_{\text{相加}}=1$ ，则将DR中的数与RA中的数相加，并将结果保存到RA中

所以很显然，要想把一大堆数加起来，只需要在存储器准备好它们，并将地址计数器AC清零，然后坐下来，以5次为单位，不停按开关K即可

合上 $K_{\text{装载}}$ ，断开 $K_{\text{相加}}$ ，按5次K；再断开 $K_{\text{装载}}$ ，合上 $K_{\text{相加}}$ ，再按5次K，这个过程中有自动的成分，毕竟，我们只是按开关，其他事情由机器完成。当然在这个过程中，唯一麻烦的就是 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 这两个开关，要在适当的时候切换一下它们的状态，要把它们省掉，该多好啊！

所以要想进一步自动化，必须把 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 省掉。但没有那么简单， $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 很重要，在图11.20中，必须不停地按开关K才能使机器工作，问题在于，同样是按K，这台机器会根据 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 的开合状态做不同的事情

换句话说，这台机器是按指令行事的，合上 $K_{\text{装载}}$ 表示下达的是装载指令，合上 $K_{\text{相加}}$ 表示下达的是相加指令。同样是按K，指令不同，工作过程也不一样。

问题是我们就是不喜欢 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 这两个开关，如何才能将它们省掉呢？办法是有的，这个问题的答案来自冯·诺依曼

约翰·冯·诺依曼1903年12月28日生于匈牙利布达佩斯，父亲是一个银行家，在欧洲一些国家“冯”是贵族的称号，比如铁血宰相冯·俾斯麦

第二次世界大战期间，由于军事上的需要，诺依曼参与了计算机方面的研究工作，主要侧重于计算机的组织形式和体系结构，战后，他开始研究自动机理论，并对自动机和人脑思维过程的特点进行了比较。作为成果，他写了一份讲稿《计算机与人脑》，可惜还没有完成就于1957年去世了。

目前为止，我们的存储器里全都是一些等待被加起来的数字。但冯·诺依曼认为，存储器里不但要有这些纯粹的数字，还应当有一些指示如何加工这些数字的指令。在《计算机与人脑》中他写道：“一条指令，在物理意义上和一个数是相同的。”换句话说，它们躺在存储器里，很像普通的二进制数，但实际上不是。就好比大喊一声“猪！”，有时，你指的是那种哼哼叫肥胖的动物，有时用来侮辱人

如图11.21所示，所有的指令都以一个操作码开始，它指示出该指令的功能。比如，用10001表示“装载”，用10010表示“相加”

地址	存储内容	具体含义
0000	10001	装载 ( $RA \leftarrow$ 下一个存储单元里的数)
0001	01010	数 (10)
0010	10010	相加 ( $RA \leftarrow RA +$ 下一个存储单元里的数)
0011	00101	数 (5)
0100	10010	相加 ( $RA \leftarrow RA +$ 下一个存储单元里的数)
0101	00111	数 (7)
0110	10010	相加 ( $RA \leftarrow RA +$ 下一个存储单元里的数)
0111	00010	数 (2)
1000	10010	相加 ( $RA \leftarrow RA +$ 下一个存储单元里的数)
1001	00110	数 (6)
1010	未使用	
⋮	⋮	
1111	未使用	

图11.21 保存了一些指令后的存储器布局

除此之外，操作码还隐含了一些别的意思，比如装载指令，往哪里装载呢？装载谁？这个数字在哪里？所以操作码10001还意味着被装载的数位于下一个存储单元里，目标是寄存器RA

相应的，相加指令的操作码10010则隐含了更多的意思。首先，它提出，第一个相加的数字位于寄存器RA；第二个相加的数位于下一个存储单元；最后，相加的结果还要保存在寄存器RA中

因此，一条完整的指令总是以操作码开始，后面跟着操作数。有的指令不需要操作数，只有操作码就行了；而有的指令很复杂，可能需要不止一个操作数

这样安排后，剩下的总是就是如何译出这些操作码，并用来代替那两个开关 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 。这好像不难，考虑到逻辑电路的输出不是0就是1，等效于开关 $K_{\text{装载}}$ 和 $K_{\text{相加}}$ 的闭合与断开，那么，使用我们非常熟悉的译码器来模拟传统的开关应该不错

接下来，我们该如何让机器读取这些指令并执行它们呢？如图11.22



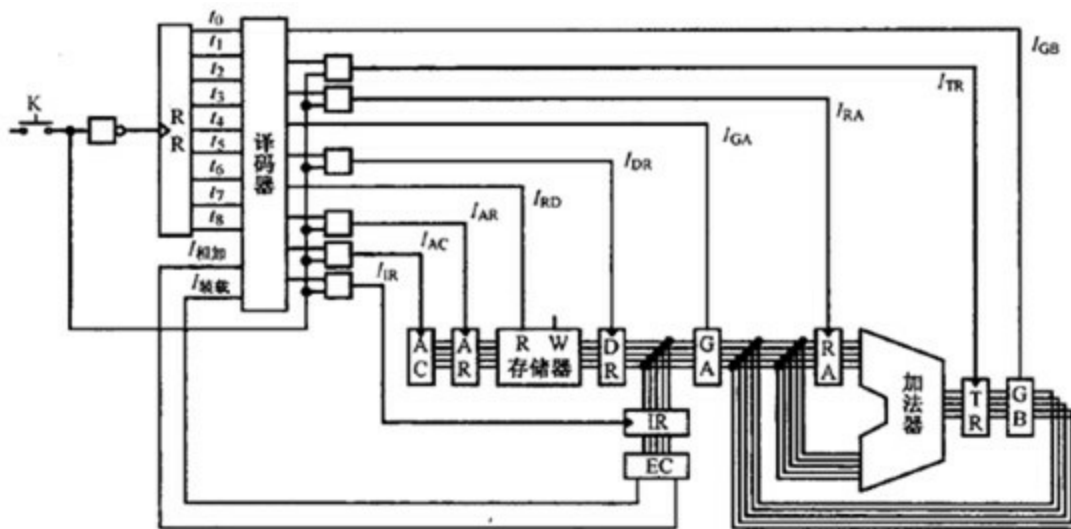


图11.22 用一只开关就可完成将所有的数从存储器依次取出并逐个相加的过程

假设这台机器刚通电启动，随着开关K的按动，在 $t_0$ - $t_2$ 阶段，第一个存储单元（地址0000）里的操作码被取出，地址计数器AC自动加一，这个阶段，称为取指令阶段

现在，第一条指令的操作码已经位于寄存器DR中，在 $t_3$ 阶段， $I_{IR}$ 产生一个上升沿，使得寄存器IR将操作码保存起来，IR是一个普通的寄存器，但专门用来临时保存指令，称为指令寄存器

IR的输出直接通向译码电路EC，EC的任务是翻译当前指令，看它到底想做什么。当它的输入为10001时， $I_{\text{装载}}=1, I_{\text{相加}}=0$ ；相反，如果输入为10010，则 $I_{\text{装载}}=0, I_{\text{相加}}=1$ ；而对于其余任何输入， $I_{\text{装载}}$ 和 $I_{\text{相加}}$ 都为0；所以， $t_3$ 阶段称为指令译码阶段

注意， $I_{\text{装载}}$ 和 $I_{\text{相加}}$ 对 $t_0$ - $t_3$ 阶段没有影响，不管它们俩输入的是是什么，机器所执行的都是取指令和翻译指令

因为第一条指令是装载指令，所以 $I_{\text{装载}}=1, I_{\text{相加}}=0$ ，于是从 $t_4$ - $t_8$ 阶段，依次执行下面的任务：从下一个存储单元里取数、地址计数器AC加一、把取出来的数装载到寄存器RA中

至此，第一条指令执行完毕，循环移位寄存器RR已经经历了一次完整的循环移位

在第二个 $t_0$ - $t_3$ 阶段，将取出第二条指令（相加指令）并进行译码，使得 $I_{\text{装载}}=0, I_{\text{相加}}=1$ ，于是在第二个 $t_4$ - $t_8$ 阶段，将再次取数，并与RA中的数相加（结果依然返回RA中）

基本上，这台机器的工作过程就是这样，可以继续按动开关K，直到所有的指令都执行完毕

## 11.5 全自动计算

在上面的例子中，我们有5个数要相加，为此需要编制5条指令，为了执行每条指令，都要按9次开关K，所以你唯一的工作就是坐下来，不停按开关。总共需要不停地按 $5 \times 9 = 45$ 次开关K

当然能够达到这一步，已经很先进，很了不起了。但如果能自动按开关呢？

按动开关只是用来发出一个由低到高、再到低的脉冲。如果连续按动，就相当于一个人肉振荡器，那么为什么不能用一个振荡器呢？

没有理由不这么做，但这同时也带来一个总是。当我们用手工来产生类似于振荡器的脉冲时，一切都是可控制的，主要是脉冲次数，如果使用振荡器，将不知道机器在工作时已经经历了多少个脉冲，当所有的数都加完后，振荡器必须停下来，但遗憾的是这些机器不像你一样有大脑，在这种情况下，这台机器将持续计算，直到地址计数器AC计数到最大，然后又接着从0000开始重新计数，当你意识到应当停止振荡器时，除了一个错误的结果外，什么也得不到

为了获得使用振荡器带来的好处而又能使它处于可控状态，需要做几个方面的改进工作

首先，必须为这台机器增加一个新的指令，即停机指令，比如11111，并把它放在其他所有指令后面。不像我们已知的其他指令，它只有操作码而没有操作数，实际上也不需要。

其次，重新设计译码电路EC，使它除了可以译出 $I_{\text{装载}}$ 和 $I_{\text{相加}}$ 外，还能译出停机，即 $I_{\text{停机}}$

最后，也是最重要的部分，重新设计这台机器的控制器，如图11.23所示



## 第12章 现代的通用计算机

在本书开头，我们的目标是制造一台计算机，但到目前为止，我们只发明了一台全自动运行的加法机。

自动加法机也是计算机，虽然它不能听音乐、看电影、玩游戏、上网.....但它有存储器、运算器（只能做加法）、奇妙的控制器、甚至包括一个用来驱动控制器的心脏 – 振荡器，所有这一切都是组成一个现代计算机所必不可少的

把指令放在存储器里，然后加以执行，这是冯·诺依曼的主意，为此很多人称他为“计算机之父”，但他说计算机的基本概念属于图灵。

阿兰·图灵（1912-1954）是英国的数学家的逻辑学家，1936年，他在24岁时，在他的一篇论文中提出了“图灵机”的理论

千万不要误会，我的意思不是说图灵写了一篇论文，阐述如何制造一台叫做“图灵机”的计算机，不是这样的。尽管图灵和诺依曼生活在同一时代，但图灵的精力更多的是花在数学和逻辑上，事实上，那篇论文说的是另一件事情，讨论的主题是逻辑的赛程性，即所有数学问题是否在原则上都是可解的。作为论证过程的一部分，图灵提出了图灵机的概念，谁也没想到，这个小小的构想会引起世人的关注，闪耀着那么夺目的光芒。

要想把图灵机是怎么回事说清楚不是件容易的事儿，得先从9世纪说起，波斯数学家阿勒·霍瓦里松和他的《代数对话录》，告诉了大家什么是算法；然后作为一个实例，古希腊欧几里德算法，通过固定的步骤来得到两个数的最大公约数；最后，近代1900年，在那一年举行的世界数学家大会上，德国数学家戴维·希尔伯特提出了一个有关逻辑完备性的问题，即是否所有的数学问题在原则上都是可解的。于是图灵写了上面那篇论文，并用一个图灵机的模型作为注解，回答了那个有关逻辑完备性的问题（即是否所有数学问题在原则上都是可解的），即有些数学问题是不可解的。

图灵机从来没有成为一台真正的机器，它是想象出来的，但却给同时代的人以启发。我们现在的计算机大部分都采用了冯·诺依曼的设计，所以称为冯·诺依曼体系结构，不可不论，冯·诺依曼的设计实际上可以看成图灵机的一个简单实现。

图灵的思想为现代计算机的设计指明了方向，现在，我们唯一要做的就是按这种思路来继续完善我们的自动加法机，来看看现代计算机大体上是怎样工作的，以及它如此有用的原因。

## 12.1 更多的计算机指令

在第11章，我们已经发明了一台能够自动运行的加法机。最重要的是，它是依靠执行指令来工作的，这就使它成了一台真正的计算机，因为现代的计算机就是按这种方式工作的，当然，它现在还很简单

指令的执行是一个有趣、巧妙的过程。正如我们已经看到的，在计算机的内部有各种各样的小东西 – 计数器、译码器、加法器、传输门、寄存器等，要使计算机能够按这些器件发出的指令工作，必须巧妙地安排连线，并预先设计好各种指令所需要的操作序列。然后，在振荡器有节律的跳动下，所有相关器件都应该在恰当的时机“动”一下。就这样，数据从一个地方出来，经过不同的器件，最后变成另一种形式，回到另一个地方（目的地）

从另一个角度来看，让计算机执行指令就像在法国用法语点菜，在英国用英语点菜，所以，把指令看作是计算机的语言或母语，是很贴切的。

从先贤们发明了计算机开始，围绕它工作的有两类人，一类人关心的是如何制造计算机的实体，也就是实实在在能看见的东西，这称为硬件；另一类人不太在意计算机内部是怎样工作的，他们只想知道硬件能执行哪些指令，并把这些指令编排到一起做某件事，这个过程称为编程，编排好的指令称为程序。这是很贴切的，在现代汉语里，“程序”的意思是事情进行的先后次序，编排指令也是一样，先用哪条指令，先计算什么，是分步骤按顺序来的。相应的，负责编排程序的人称为“程序员”

在第11章，我们讲了全自动加法计算机，从程序员的角度来看，这台机器的内部构造是次要的，在他们眼里，这台机器只有三样东西有价值，第一样，是存储器，因为他们需要把指令和数据存储在这里；第二样，是加法器，因为它是数学计算实际进行的地方；最后一样对程序员来说尤其重要，它就是寄存器RA，原因很简单，程序员可以不知道指令是，但绝对要知道机器把执行的结果放到哪里了

如图12.1所示，这是程序员眼中的计算机，很显然，尽管在计算机的内部还有很多寄存器，但它们是临时寄存器，只有RA。造成这种差别的是身份问题：谁有资格出现在计算机指令中

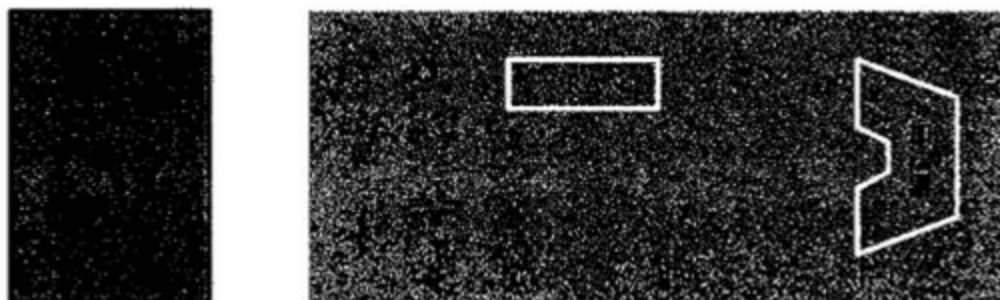


图12.1 我们已知的两条指令只和寄存器RA打交道，  
对程序员来说，计算的其他内部细节并不重要

比如，我们已经两条和数学计算相关的指令，即装载和相加，它们都需要寄存器RA，对于装载指令来说，它只是需要RA中有一个我们想要的数字；而对于相加指令来说，RA既是两个相加的数字之一，又是加法结果的归宿，否则，不知道计算机把结果放在什么地方，所以就没有办法找到它，这不是计算机的义务

作为一个例子，我们把装载指令的操作码定为10001，它指示了三层意思：第一，这是个装载数字的指令；第二，目的地是寄存器RA；第三，该指令还包括一个操作数，它位于操作码后面的下一个存储单元

所以，不要小看这个“二进制数”，它不但让你知道要做什么，结果到哪里去找，还包含了足够的信息让计算机开始干活儿。如图12.2所示，第一条指令的操作码为10001，操作数是00110，表示要把数字00110装载到寄存器RA中

一旦执行这条指令，寄存器RA里的数字就是00110，也就是十进制的6，换句话说，当这条指令执行的时候，装载到寄存器RA中的数字直接来自指令本身，是该指令的组成部分，可以从指令中立即得到，所以称为立即数

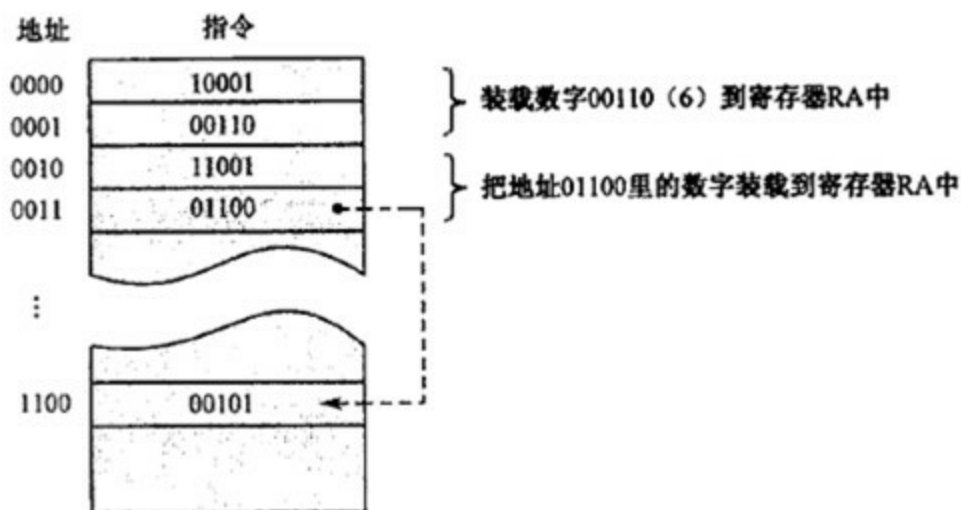


图12.2 两种不同的装载指令，它们的工作方式大相径庭

在其他书籍里，装载指令称为传送指令。不管是装载还是传送，都

未能充分表达指令本身的功能，因为东西一旦传递出去，就少了一件，而指令所做却是在目的地复制一份。为了方便地书写指令，传送（装载）指令需要一个简单易行的表示方法

**MOV RA,16**

MOV是MOVE的缩写，本身就有传送的意思，这条指令表示把立即16传送到寄存器RA中，既然这种写法是写给我们人类看的，就没有必要采用二进制，十进制更直观

既然有立即数，那么肯定有“非立即数”、“间接数”……，确实有这样的情况，指令要操作的数字并非来自指令本身，但指令却告诉我们到哪里能找到这个数字

如图12.2所示，同样是装载数字的指令，但它却具有另一个不同的操作码11001，这是因为它需要向控制器表达另外三层意思：第一，这是个装载指令；第二，目的地是寄存器RA；第三，跟在操作码后面的不是一个立即数，而是另一个存储单元的地址

因为这条指令的操作数是01100，所以它指向存储器的另一个地址1100（十进制的12），而这个地址里存放的是数字00101，也就是十进制的5，所以当这条指令执行完毕时，寄存器RA中的数字是5，与第11间里那台简单的自动加法机不同，我们需要添加额外的硬件来把该指令的操作数作为地址赋给地址寄存器AR，并再次访问存储器以取得实际数字

尽管同样是装载指令，为了表明它的操作是一个地址，需要另外一个标志，比如一对括号：

**MOV RA,(2)**

通常情况下，计算机的存储器容量足够大，不但可以存放程序指令，还可以留出一部分来存储数据，因此，如果我们在编写程序时，不在指令中使用立即数，而代之以间接的地址，就不用把程序改来改去，或重新编排，只需要每次把那个地址里换成不同的数字就可以了

存储器有足够的空间，一旦我们决定要开垦它，就会发现这里需要更多的指令（以及更复杂的硬件设计）。比如，可以将寄存器RA中的数字传送到某个存储单元里，当然，需要在指令中指定一个地址：

**MOV (25),RA**

执行这条指令，将把RA中的数字传送到地址为25的存储单元里，甚至可以直接把一个立即数传送到指定的存储单元里：

**MOV (30),20**

尽管表面看不出来，但这条指令是我们所见过的最复杂的，因为在操作码后跟了两个操作数，一个是存储单元地址，另一个是立即数

最后关于传送（装载）指令我们要说的是，有些计算机不允许在两

个存储单元之间传送数据，所以在那种计算机上，像这样的指令是不存在的：

**MOV (23),(18)**

理由很简单，在两个存储单元之间传送数据可以用寄存器中转（即使你不这样做，计算机在内部也会用临时寄存器来中转），这样可以节省硬件成本并减小设计难度

说完了传送（装载）指令，再来说说相加指令，我们实际上已经有一条这样的指令了，在那里，寄存器RA中的数字和指令中的立即数（比如25）相加，结果返回RA中，这条指令可以表示成：

**ADD RA,25**

ADD的意思是把一个数和另一个数相加，在指令中直接指定立即数不是个好主意，因为一旦数字变了，就要重写指令，为了获得一些灵活性，有必要将相加操作和存储单元建立关联，比如，可以用某个存储单元里的数字同寄存器RA中的数字相加：

**ADD RA,(12)**

指令执行完毕，寄存器RA中将获得相加的结果，不过要是想把结果放到存储单元里，换一条指令：

**ADD (12),RA**

像寄存器一样，存储单元也可以直接和指令中的立即数相加，当然，需要另一条指令：

**ADD (22),9**

在某些类型的计算机上，同样不允许两个存储单元之间直接做加法和存储器有关的指令使我们得以充分利用它的空间。比如，为了将任意两个数相加而不用重写程序，我们编写如图12.3那样的指令



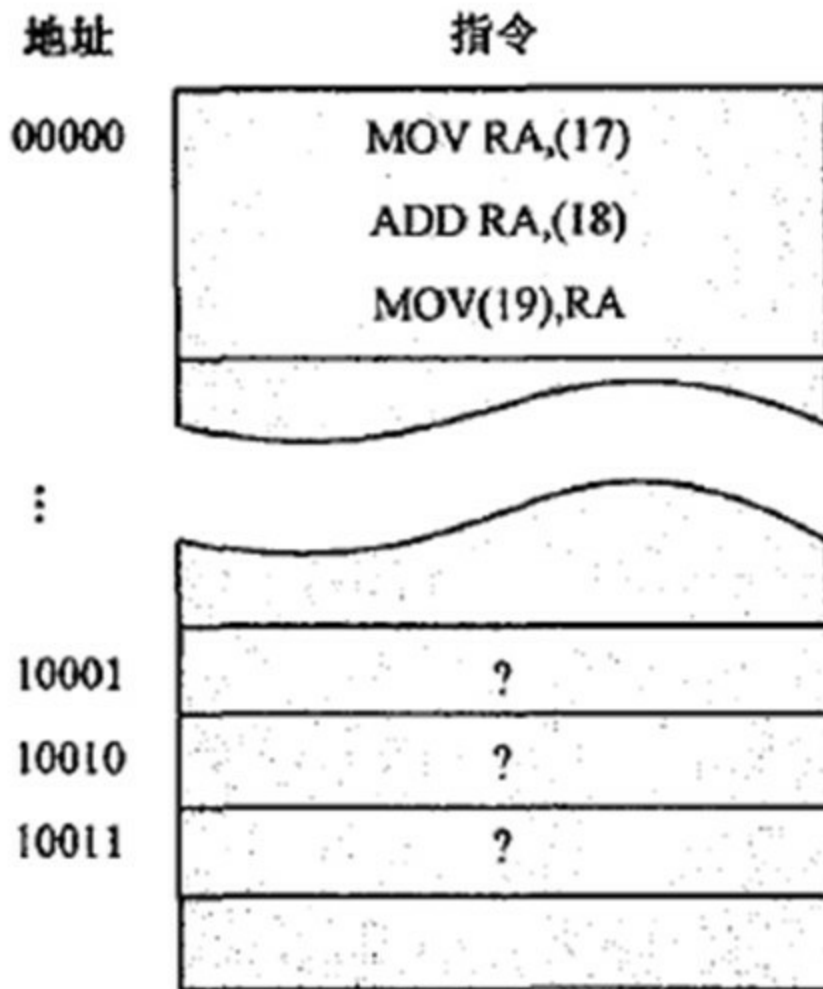


图12.3 通过在指令中使用地址，即使参与计算的数字变了，也不用重新编写程序

稍费点心思，这几条指令的意思是不难理解的，尤其是当你知道二进制数代表什么十进制的时候。很明显，程序写好之后就不用再更改了，每当你有两个数字相加时，就把它分别放在地址为17和18的存储单元里然后让机器从头开始执行，完毕后就能在地址19的存储单元中得到结果

## 12.2 当计算机面临选择时

不知道你发现了没有，我们在使用加法指令时，有一个潜藏的问题或者说隐患。是什么呢？看下面的指令

**ADD (22),RA**

一般的，存储器的每个存储单元和寄存器具有相同的比特数 – 换句话说，数据宽度。考虑到我们一直把存储器和寄存器设计成能读/写5个比特，所以现在延续这种做法。

5个比特意味着，每个存储单元，及寄存器RA，所能表示的二进制数最大的为11111，也就是十进制的31

看上面那条指令，假如地址为22的存储单元里保存的是11001（十进制25），寄存器RA中保存的是10110（十进制22），那么执行这条指令后，结果就应该是6比特的101111（十进制47），遗憾的是，为了同存储器和寄存器取得一致，加法器的输入线也是5根，在这种情况下，加法器输出的是01111，并在内部产生一个进位，所以，当那条指令执行完毕后，地址为22的存储单元里实际上得到的是01111

这当然不是正确的结果，但眼下我们无能为力，唯一的办法就是重新设计这台计算机，并添加一些新的指令，这些指令可以判断是否产生了进位，然后分别做不同的处理。

比如，我们可以用两个存储单元来保存相加的结果，反正存储器容量很大，如图12.4所示

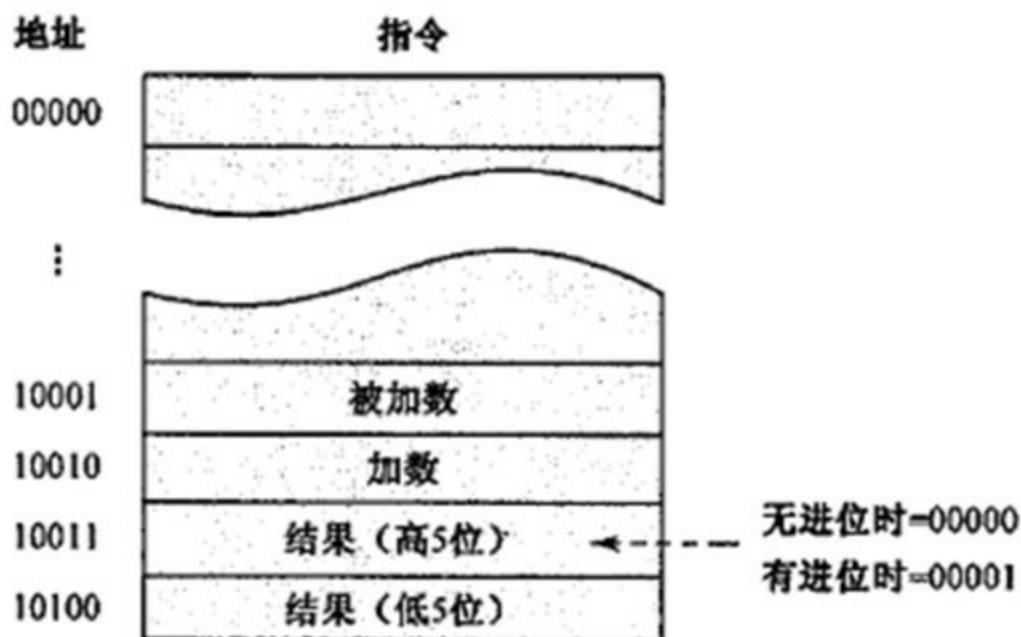


图12.4 两个5位的二进制数相加，结果可能是6位，

所以必须分配2个存储单元

存储器地址10001（17号存储单元）和10010（18号存储单元）里保存的分别是两个要加的数；地址10100（20号存储单元）用于保存相加的结果（右5位也称低5位）。如果相加时产生了进位，则将数字0传送到地址10011（19号存储单元），否则就传送一个1，这样，当需要显示或打印时，将两个存储单元组合就是正确的计算结果

问题是进位是不可预知的，有些数相加会有，而另一些则没有，该如何在计算机运行的时候进行干预或者观察到呢？

不需要这样，阿兰·图灵和冯·诺依曼早给出了答案

在图灵和诺依曼关于运算机器的思想里，很重要的一点被我们忽略了，那就是每条指令的执行不仅取决于这条指令的目的和功能，还取决于上一条指令执行的结果。比如说，一个聪明的猎人每天可能会“执行”两条“指令”

听天气预报

打猎去

取决于天气预报的结果，如果下雨，那么打猎指令的执行结果就和平常不一样

同样的道理，将17、18号存储单元里的数字相加需要以下指令：

MOV RA,(17)

MOV (2),RA

MOV RA,(18)

ADD (2),RA

很好，计算的结果的低5位已经位于第20号存储单元，现在要做的是，就是根据上面最后一条指令：

ADD (2),RA

判断是否产生了进位来进行不同的处理，如果没有产生进位，就把19号存储单元写0

MOV (19),0

否则就写1

MOV (19),1

我们知道，指令在存储器里的存放是一条接一条的、按顺序来的，而执行的时候也是这样。这就意味着，要解决目前的两难问题，需要一条指令，能根据是否产生了进位来跳转到不同的存储器位置接着执行。

现在回到第6章，看看图6.4，两个3比特的数相加，结果由4根线引出，其中 $S_3$ 是最后一个全加器的进位输出，同样的道理，在我们的加法器里，两个5比特的数相加，应该产生6比特的输出，但第6个比特，也就是最后的那个进位被扔掉了，或者悬空了，没有使用

这当然是不行的，如图12.5所示，我们把加法器的进位线拉出来，

接到一个D触发器上，这样，每次计算结果出来的时候，如果没有进位，则Q=0；否则Q=1

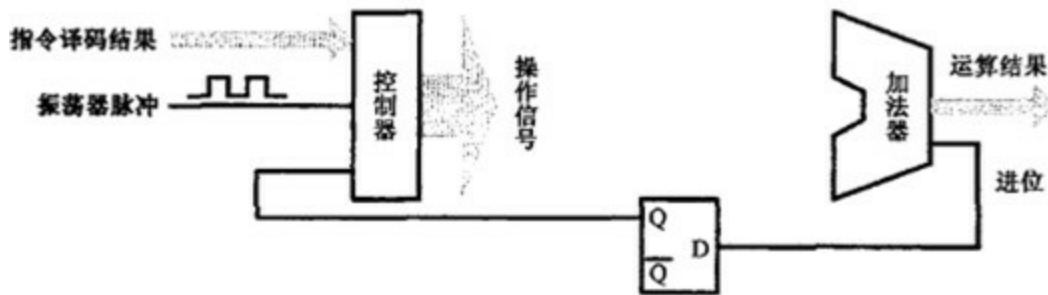


图12.5 上一条指令的执行状态被送到控制器，将改变后续指令的操作信号

加法器的计算结果有可能会很快消失，但触发器上的进位却能一直保存，直到下一条指令的结果把它挤掉

从图中可以看出，触发器的Q通往整台计算机的控制器，对于控制器来说，这个输入称为进位标志，它和指令译码结果一起，可能改变很多指令的执行过程，如果它们“愿意”被它影响的话

那么，这正是我们希望的，因为我们根据是否产生进位来跳到其他指令那里去执行，比如：

JC 51

JC是Jump if Carry的缩写，意思是如何进位则跳转，所以，一旦前面的指令产生了进位，这条指令将使计算机跳到存储器地址为51的地方接着往下执行。但，如果进位实际上没有产生，那这条指令执行后什么也不会发生

如图12.6所示，一旦指令

ADD (20),RA

没有产生进位，那么接下来的指令

JC 12

将什么也不做，计算机接着执行

MOV (19),0

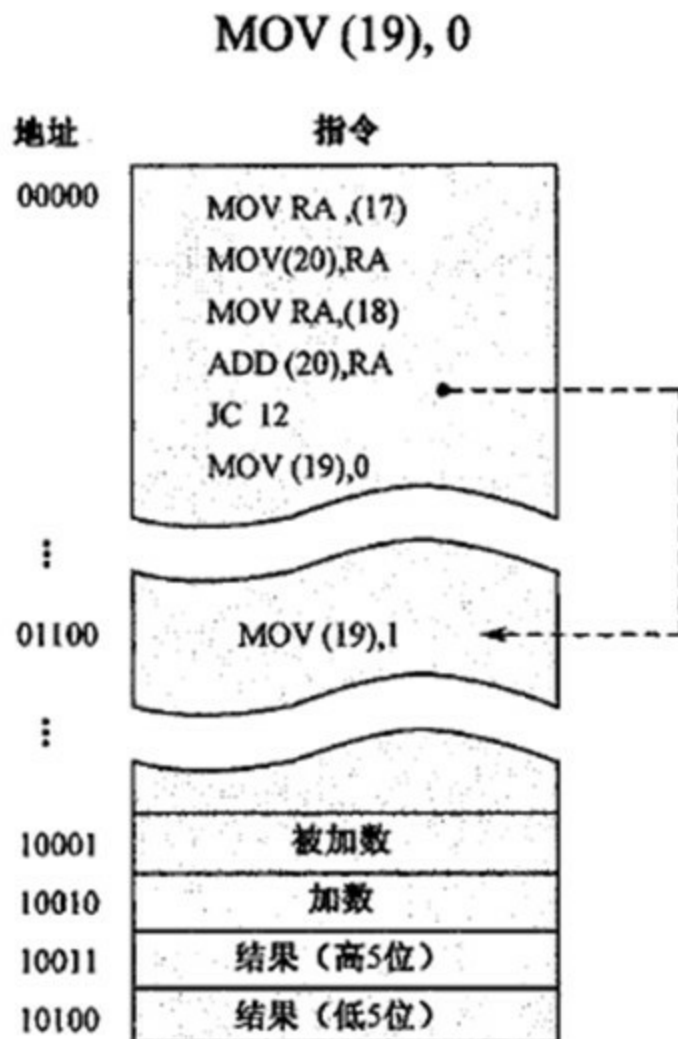


图12.6 跳转指令示意图

但，如果有进位产生，那么跳转行为将实际发生，计算机跳转到存储地址12（即01100）那里执行，执行的指令是

MOV (19),1

看得出，为了解决进位问题，我们兜的圈子真不小。实际上，两个数相加没有这么麻烦，很多计算机都设计了一条带进位加法指令，比如：

ADC RA,30

这条指令的执行取决于前一条指令，如果前一条指令没有产生进位，那么它就是单纯的加法，和ADD指令没有任何区别；相反，如果前一条指令产生了进位，那么，它将RA的内容和30相加之后，还要再进位1

所以，前面的进位难题可以很简单地这样来解决：

MOV (19),0  
MOV RA,(17)

```
ADD RA,(18)
ADC (19),0
MOV (20),RA
```

## 12.3 现代计算机的大体特征

一般来说，存储器、寄存器和加法器具有相同的数据宽度 – 它们的数据引线具有相同的条数。比如说，如果数据线有8根，则存储器的存储单元包含8比特，寄存器RA需要用8个上升沿D触发器来制造，而加法器则必定由8个全加器组合而成，如图12.7所示。换言之，招集亲朋好友开宴会，有多少个人，就得准备多少套餐具，这个称为计算机的字长

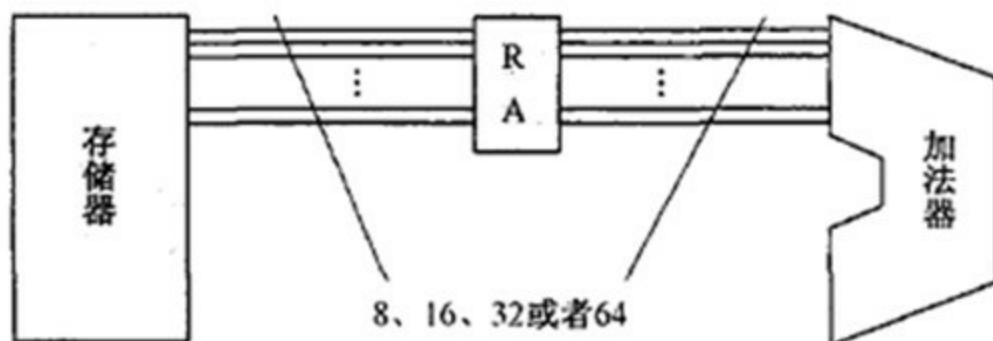


图12.7 字长

字长表示一台计算机在一次操作中可以处理的二进制比特数，换句话说，就是每个寄存器可以保存的二进制数是几个比特，或者加法器每次计算的二进制数是几个比特，原则上，对计算机的字长没有任何规定的限制，4、6、20比特等等都是可以的。

鲁迅说“其实地上本没有路，走的人多了，也变成了路。”20世纪50年代，一个计算机公司决定把它的产品设计成8位的字长，这是有原因的，一个二进制数，它可能不单是一个真正的二进制数字，也可能用于代替其他事物，比如26个英文字母、阿拉伯数字、标点符号，以及用于向打印机这样的外部设备发送控制命令（如果不是这样的话，你就不能用计算机写报告并把它打印出来，你所写的每一个字，包括标点符号，在计算机内部都和二进制数字一模一样）。经过仔细考虑，最终，他们认为使用8位的二进制数就足以包括这一些 – 不太多，但又刚好够用。

8位的字长称为字节，由于那种计算机的畅销，这事实上已经成为一个“标准”，最开始的时候，他们把字节拼写成bite，新鲜了没多久，可能觉得它容易和已经流行很广的bit混淆，于是改成了byte

传统上，说到“字节”时，大家习惯用一个大大的字母B表示，以区别表示“位”的小写b，所以

$$1B=8b$$

$$1\text{Byte}=8\text{bit}$$

$$1\text{字节}=8\text{位}$$

字长是计算机的一个重要技术指标，也许还是最重要的一个指标，要是字长太小，就表明这台计算机每次只能计算很小的数，例如，对于一个8位的计算机来说，它的被加数和加数不能超过二进制的11111111（255），这还没有考虑相加后结果可能超过8位，这将使寄存器放不下

不过，这并不意味着一台8位的计算机就不能计算像1050+7000这样的数学题，只是有些麻烦，需要分好几次进行计算（使用ADD和ADC指令）

但如果我们用一台字长为16位的计算机来做这道题，则只一次就可完成。重要的还不是计算过程，因为涉及访问存储器，两次访问存储器比只访问一次要花费成倍的时间，这就好比每次只能扛50斤的东西，150斤的东西得分三次扛走，相反要是你一次能扛150斤的东西，扛一次就够了

对于我们平时所用的个人计算机来说，几十年前8位是主流（也有4位的计算机），后来是16位，现在的计算机部分是32位，还有64位的。

字长是一个复杂的问题，不是一次可计算多大的数字那么简单，想想看，在8位计算机的时代，一条装载指令将从存储器取得一个8比特的操作数；而在32位计算机上，必须让这条指令仍以8位的模式工作（这称为向后兼容性，兼容性是必须考虑的问题，为了保证已有投资，那些老的程序必须继续发挥应用的作用，重新编写这些程序需要花钱），同时，专门为新计算机而高见远识的指令则需按32位的方式操作，以获得升级换代的好处。在这种情况下，该怎样设计存储器和寄存器，并如何连线，以使它们井井有条、和谐共存呢？

遗憾的是要解决这些问题，本书无能为力，不过，这并不是说无法解决，相反，解决得很好，无非是修修补补。就像我们很希望城市里全是整齐划一的街道、建筑，但实际上通常都是街道弯弯曲曲、错综复杂、建筑高低不同，但整个城市还是运转得不错

除了字长的区别，一台计算机只能计算加法也是个问题。所以，我们应当让计算机能够做诸如减法、乘法等这些运算。因此，在造出了加法器后，再接着制造减法器、乘法器和除法器等等。这对于已经掌握了数字逻辑知识的人来说，并非难事，但已经不是本书要关心的事情了

你也许希望我们把幂、开方等这些功能也纳入进来，但没必要，因为数学的基本任务是将复杂的、高等的运算转化成基本的加、减、乘、除四则运算。比如5的平方实际上可以转化成两个5的乘法

$$5^2=5*5$$

所以这台机器只要具备加、减、乘、除这四种基本运算功能就足够了。通常，这几种运算功能的电路结合在一起，称为运算器

除了算术计算功能的增强，现代计算机也会增加可以在指令中使用



的寄存器数量。比如说，可以在RA的基础上，继续增加寄存器RB，RC和RD，如图12.8所示



图12.8 现代计算机的内部会有不止一个寄存器

可能觉得4个寄存器还是太少，寄存器多了虽然好，但这是以增加计算机内部电路的复杂度为代价的，需要更多的连线、更多的传输门和更复杂的控制器，制造成本也会迅速增加，所以，现代的计算机都只有少量的寄存器供使用

为了用好这些资源，现在唯一要做的就是添加指令，首先，所有的寄存器，以及任何一个存储单元，都可以装载一个包含在指令中的立即数

其次，任意两个寄存器之间或寄存器和存储单元之间，都可以互相装载数据。比如，可以有一条指令将寄存器RC中的内容装载（复制）到RA中；再比如，可以把寄存器RB中的内容装载（复制）到存储器中地址为255的存储单元中

再次，所有的寄存器以及任何一个存储单元，都可以和包含在指令中的立即数进行加、减、乘、除运算，结果依然返回该寄存器或存储单元。比如，我们前面已经熟悉的相加指令

最后，寄存器之间，或者寄存器和存储单元之间，都可以互相进行数学运算。

除此之外，现代计算机还会根据一条指令的执行情况产生各种标志，比如我们已经熟悉的进位标志。其他的标志还包括计算结果为0，结果中1的个数为奇数/偶数等等。这些标志可用于跳转指令，或者其他想要参考这些标志的指令

可以想象，对于现今的任何一台计算机来说，都需要大量的指令支撑它们的运转，用于解决我们所可能碰到的方方面面的问题（但不可能是所有问题，比如，吃饭）。对于任何一种类型的计算机来说，它的指令在种类和数量上都是有限的，但不管有多少，它所能执行的所有指令，称为这种计算机的指令集

## 12.4 为什么计算机如此有用

更多的指令和更强有力的控制器能让计算机做越来越多的事情。但

无论它的功能有多少，我们仍然没有摆脱一个尴尬处境，那就是，用机器算数学题更费劲，还没有手工做这些事情来得快，想想看，比如下面这道数学题：

$$(10+3\times 6-5)\div 5$$

首先，就像我们一直喋喋不休地强调的那样，你不可能指望机器具有自主意识，能够听懂你对它说“题都写在这张纸上了，好好给我算着！”

在这种情况下，你唯一所能做的，就是坐下来老实地编排指令。这个过程完全和你手工做这道题一样，要考虑先做乘除，再做加减、谁和谁相乘，然后再用结果和谁相加等等。毫不夸张地说，如果一道题连你都不会做，那么就不要指望机器会做。毕竟，你必须给出详细的计算步骤，像称职的保姆那样安排好一切，机器所做的，只不过是执行你设计好的步骤

说到这时，相邻你已经明白我的意思了，这是一道非常简单的四由混合运算题，直接心算很快就能得出结果，要是用计算机来算，首先得编排程序指令，然后一个一个写入存储器，拉下闸刀，让机器开始执行，这即使不是一个漫长的过程，也够让人厌烦的。难道这就是我们发明计算机所带来的好处？

遗憾的是，对于这个问题的解答可能不会使你感到高兴。用计算机来算题，你得把它“侍候”到位。必须用你自己的解题过程来编排程序指令，并写入存储器，这个过程甚至比你自己心算、笔算来算题还要麻烦，这是事实，即使是今天，也是如此。

不过，看到当今世界范围内的计算机产业如此红火，人们都以会使用计算机而感到兴奋的时候，你也许就不那么沮丧了，但你依然不太明白，为什么会发生这种神奇的事情呢？

首先，我们的一生是发现规律、按规律生活的一生，这就是我们存在的方式。每隔一段时间你要洗澡，因为你知道不洗澡身上会痒，这就是规律；科学家的职责是发现了规律；同样有人夸奖你业务熟练，只能说明你也已经发现了规律。最后，计算机之所以有用，仅仅是因为我们只让它干有规律的事情，这里有一个例子，比如计算两个数之和的平方

$$(5+9)^2$$

$$(3+5)^2$$

$$(7+6)^2$$

如果就事论事，重复地为这三道题各自编排指令，这当然是很麻烦的，更不要说以后还会遇到类似的题目

事实上，我们都知道，两个数和的平方对应着一个公式：

$$(a+b)^2=a^2+2ab+b^2$$

即使我们的算术逻辑单元不能计算平方（这并不奇怪，很多计算机不提供这种运算）也没有关系，因为上面的式子可继续展开

$$(a+b)^2=a^2+2ab+b^2=a*a+2*a*b+b*b$$

为了获得灵活性，我们在存储器里专门开辟了几个存储单元，用于存放这两个数a,b，以及计算结果。即使你不知道a和b是多少，也没有关系，因为我们的程序将从这些地址里取得实际的数字，然后进行计算

一旦这些指令编排完毕，无论你想计算哪两个数的和和平方，所要做的仅仅是把那两个数字分别写入存储器的固定位置，然后命令机器开始计算，而不用重复编写程序。一次编写，重复使用，这就是你能获得的好处。当然，对于解释计算机为什么有用，这个例子太微不足道了，不够份量，不过我们的目的是说明问题。

我们都知道圆周率 $\pi=3.14159\dots$ 这是一个无限小数，据说它的小数部分已借助计算机算到了万亿位，也有很多人热衷于通过背诵它来彰显自己非凡的记忆力。但因为它太长，而且这种比赛的组织者显然很不通情理，要求背诵的时候不能停顿，所以选手们得穿着尿不湿才行

当然，背诵圆周率和穿不穿尿不湿，对于本书来说不太重要，重要的是这个无穷无尽的圆周率，可以用简单的加减乘除来进行推算。这个推算的过程，和前面那道题一样，也是几个简单步骤的无数次重复，用机器来做很快就能得出结果。要是用人来算的话，不知道要算到猴年马月，也许几代人的时间加到一起也算不出来

再比如说你用计算机听歌，尽管你每次播放的歌曲不同，但在计算机的内部，播放器总是在按已经设定好的、相同的方式进行播放，在气象部门里，要推算最近几天的天气情况，需要借助于数学工具，也就是说，天气是“算”出来的，气象工作者称之为“数值预报”，即数值天气预报和数理统计预报，它们都是一些复杂的数学方程式，涉及大量的数据和算术运算。这个计算过程每次都一样，唯一不同的是每次参与运算的数据是不同的，这很好理解，因为每天的气压、温度、湿度、风力、风向和云层数据都是不一样的，以前靠手工计算，往往需要很长时间，好不容易算出来了，那天已经过去了，天气预报也就成了“天气后报”，现在好了，借助计算机，很快就能知道结果

所以，认为我们现在的计算机都非常智能，而且非常聪明，那只是一种错觉，真实的情况是，所有的步骤都是已经事先安排好的，而要让它干的事情也都事先经过了安排，否则它不会知道如何应付。对此，一个简单的例子就是当你在计算机上写文章时，如果输入了一个错字，想删掉它，可以按退格键，如果按了其他键，则不会达到预期的删除效果，因为在当初编写能够让你写文章的指令时，就是这么安排的，你要么照规矩执行，要么给自己带来麻烦

看起来我们所要做的是制造一台新的、能够以不变应万变的、计算各种复杂数学题的机器，当然，还得是自动地干这种事情。像这样的机器，因为它能应付各种各样的运算，所以被称为通用计算机。

## 第13章 集成电路时代

历史是纷繁复杂的，科学却要分门别类，但如果你看得仔细些，各门学科之间都是互相借鉴、互相学习的，就这样向前发展。

一开始，电学在磨磨蹭蹭地向前走，当然是越来越快，于是电磁铁发明了，也有了继电器，这个时候，电子计算机的先驱们也正处在彷徨中，看到了继电器觉得这东西挺好，都是合用的东西，可以拿过来使用。基本上，在20世纪30年代，他们都是在用继电器制造那些最原始的计算机。他们造的计算机，有的非常庞大，用了数不清有多少继电器，工作起来开关的断开闭合噼噼啪啪，声势雄壮，煞是热闹，那阵势、那场面，据亲临现场的人说“像是挤满了一屋子纺织女工”。

在没有多少东西可供选择的年代，电子管和晶体管是最理想的选择。但它们有自己的缺陷，指望用这两样东西来制造一台你面前的计算机，是不可能的。好在我们只是重温历史，现在当你坐下来使用计算机时，大概会愿意想一想，究竟是发生了什么翻天覆地的变化，才使得我们现在的计算机变得如此紧凑而轻巧呢？它都经历了哪些艰难曲折的演变过程？

## 13.1 电子管和晶体管时代

电子计算机用上继电器是在20世纪30年代，那个时候，电视机动量算子已经有了，但电子计算机的研究刚刚获得突破。要是再早些，当弗莱明发明电子二极管、福雷斯特发明电子三极管的时候，这方面的进展就更别提了，连萌芽都谈不上，完全是一片沉寂。唯一的例外是触发器的发明，那是1918年。尽管这项技术在十几年后为计算机的发展带来了福音，但这并不是发明者当初的愿望。

时间到了20世纪30年代，当电子三极管由于制造工艺的成熟和价格的降低，其应用开始爆发的时候，香农也已经完成了把布尔的数字逻辑系统与继电器相结合的工作。正如我们现在已经知道的那样，当时已经出现了为数不少的继电器计算机。当然，这都是一个一个的“小玩具”，要让它们真正变得实用，能解决复杂的总是，而且速度要快，就必须使用电子管

到20世纪40年代，程序存储 – 也就是把程序放在存储器里，由计算机自动执行的思想已经开始为越来越多的人所接受。另一方面，人们也注意到，继电器的速度很慢，而电子管则比它快千万倍。使用电子管，不但可以产生频率很高的时钟脉冲供计算机内部顺序控制之用，而且用电子管制成的与、或、非逻辑电路也能以极快的速度工作。但是，制造这样一台可以存储程序并自动进行计算的机器并不仅仅是胆识、还有钱。

一般来说，商人、政府和军队有钱，只要劝说他们把钱拿出来，得有充分的证据表明制造一台电子计算机将会获得几倍甚至更多的回报。

1943年，第二次世界大战正打的不可开交，美国军方需要为他们的新火炮制作弹道表，为了赶时间，这一次他们很痛快地斥巨资要科学家们帮助制造一台计算机。

这台机器每秒钟能做5000次加法，用了成千上万的电子管和继电器，耗时3年，当它好不容易完成时，战争已经结束了

懂得电学的人都知道，继电器和电子管的个头都不小，而且都不是省油的灯，所以由于用了大量电子管和继电器，这台机器不但在体积上大得惊人，耗电量也同样大的惊人。据说当时只能在夜深人静的时候把它打开，要不然当地居民家里的电灯都会变得黯淡无光

一台计算机应当包括一个存储器和一个运算器，指令和数据都放在存储器里，在控制器的指挥下一条指令一条指令地自动执行，这种安排正是从这个时候开始的，一直沿用到现在。由于第一个提出采用这种结构形式并积极把这种设想付诸实施的是数学家、计算机专家、美籍匈牙利

利人冯·诺依曼，所以这种结构形式又称为“冯·诺依曼体系结构”，当然，这不是唯一的计算机体系结构，建议大家了解一下什么是哈佛体系结构。

在那个时代，计算机庞大、昂贵、操作复杂，需要在大堆专家才能侍候得了，像这样一种东西，所有生活在那个时代的人都会很自然地觉得它应当被放在神殿里，只有那些复杂和重要的数学总是才配在这样的机器上算一下。这是一种在诞生的时候离普通人的生活过于遥远的东西，人们只希望它应当越来越强大，越来越快，但普通人用不上的东西，造那么多有什么用呢？难怪同时代著名的科幻作家阿西莫夫也这样预言：“一台计算机最终会有几十亿个电子管，有一个国家那么大。”不过，阿西莫夫不应当为没看到这么巨大的计算机而感到遗憾，因为不单是他，连我们都没见过

所幸的是，正如我们已经知道的那样，晶体管适时地被发明出来了，此时，电子计算机的相关理论已经相当完备。现在，它只要有一个强大的推进器。这也是第一次它能够用最短的时间搭上其他物理学发明的便车

世界上第一台晶体管计算机诞生于肖克利获得诺贝尔奖的那一年，即1956年。领先一步的工程师们有幸参与其中，目睹了采用晶体管的电子计算机成功减肥，即不需要灯丝，也不需要高压电，工作稳定，还不像从前那样费电，连续工作时间大大延长，功能也增强了，只有新的技术才能缩小体积、降低成本，当耗电量大幅度减少，计算机的体积也不再是科学家的心理负担时，可以放心大胆地设计出更多新功能的电路。

说来也巧同样是在1956年，周恩来总理亲自提议、主持、制定我国《十二年科学技术发展规划》，选定了“计算机技术、半导体技术、无线电电子学、自动学和远距离操纵技术”作为“发展规划”的四项紧急措施，并制订了计算机科研、生产、教育发展计划。同年8月25日，我国第一个计算技术研究机构－中科院计算技术研究所筹备委员会成立，主任就是数学家华罗庚，这时就是我国计算技术研究的摇篮。

在苏联的帮助下，1958年8月1日，我国第一台小型电子计算机诞生，编号103，字长32位，每秒运算30次

电子管的发明使制造一台真正的电子计算机成为可能，而晶体管则使它快速发展。不过，这两样东西还是无法使电子计算机全副武装，实际上还差得远呢。可以用它们来制造运算器和寄存器，一是没办法，二是需要不了多少寄存器这样的东西，但没有人舍得用电子管和晶体管来制造存储器，至于用继电器来制造存储器，那更是历史上从来没有发生过的事情

我们已经知道，1个字节（B）包含9个比特（b），要计量二进制数

据，字节是最基本的单位，比字节还大的是千字节（KB）

$$1\text{KB}=1024\text{B}$$

比千字节更大的是兆字节（MB）、吉字节（GB）和太字节（TB），它们之间的换算关系是：

$$1\text{MB}=1024\text{KB}$$

$$1\text{GB}=1024\text{MB}$$

$$1\text{TB}=1024\text{GB}$$

一般的，保存1个比特的成本相当于好几个电子管或晶体管，而1个字节则需要几十个。算下来，1KB的存储器就需要好几万个。对于MP3歌曲，一首歌平均3MB的数据量，100首就是300MB，要在以前，做成这么大容量的存储器需要5000000000只电子管或晶体管。



## 13.2 集成电路时代

在翻过了肖克利这一页后，晶体管没有停滞，而是以更快的速度改变着世界。我们已经说过，晶体管实际上可以做得很小，小到肉眼都看不到。但如果没有其他物理化学工艺的支持，实现不了。但因陋就简，来个简单的也不是不可能。

1958年，也许是受够了在一大堆晶体管里连接杂乱无章的导线，一个叫杰克·基尔比的美国人决心要做些什么来改变这一切。他想，一个大的电路要使用很多零件，比如晶体管和电阻这些东西，如果换一个视角来观察这个电路的工作，不过是电流从一块掺杂的硅里出来，经过导线后，又流入另一块掺杂的硅里，本质上就这么简单，为什么不把连接线去掉，让电流直接从一个掺杂区域流到另一个掺杂区域呢？

这有点儿像几个人住的很分散，要到别人家串门，得走一阵子，现在好了，他们现在搬到了一个有好多个房间的大房子里，要到别人家去只需要从一个房间到另一个房间。

就这样基尔比发明了集成电路（也就是IC，Integrated Circuit（集成电路）的缩写。一个具有某种功能的集成电路也叫芯片）。这世界上第一块集成电路，是一个振荡器，里面包含的零件不到十个。2000年，在距离他发明第一块集成电路42年后，他获得了诺贝尔物理学奖。

1959年，罗伯特·诺伊斯发明了一种新的工艺，可以在一块本征硅上制造大量晶体管，他是肖克利的学生，早年因为仰慕肖克利而在肖克利手下工作，后来因为无法忍受肖克利而离开。诺伊斯的新工艺完全建立在一套工艺流程上，具备在工厂流水线上批量生产的条件，这在当时是很了不起的。

从诺伊斯发明这种工业化生产集成电路的方法开始，在随后的几十年内，这种技术改进了很多回，每改进一回，集成的晶体管数量都会千百倍地增加。刚开始的时候，指甲大小的硅片上可集成几十个晶体管，到现在，这个数量可达几千万甚至更多。图13.1显示了两种常见的集成电路

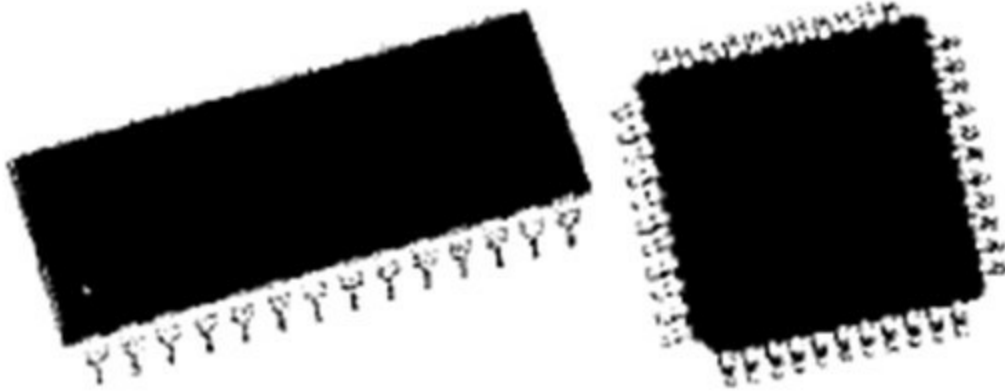


图13.1 两种常见的集成电路外观

比起晶体管，集成电路更小，更便于使用，而且耗电量更低，这是它具有光明前途的优良特征。它应该被迅速应用到电子计算机上，不是吗？

我们知道，磁芯曾经是制造存储器的主要材料，但集成电路的特点使得它很适合制造半导体存储器，于是一大批半导体存储器制造厂商诞生了，并很快终结了磁芯长达二十年的应用历史，一直到现在。以前，用5000000000只晶体管做一个大容量存储器还是梦想，现在却可以很轻松地把它做在指甲大小的一块硅片上。和以前一样，晶体管是构建触发器的材料，然后大量的触发器又可以形成大容量的存储器。传统上，这就是静态存储器（Static Random Access Memory, SRAM）的制造方法。至于磁芯，除了博物馆已经见不到它们的踪影了。唯一的例外是英语单词core，过去指磁芯，现在仍然被用来指代存储器。现代的计算机经常会出现一些小的故障，如果在偶然的情况下，你看到显示屏上出现“Exception encountered: core dump”时，除了意识到自己遇到了麻烦，或许还能感受到磁芯曾经给这个世界带来的影响

触发器的工作速度很快，所以静态存储器一直是制造计算机存储器的首选。遗憾的是，在那个时候，采用触发器来构建大容量存储器需要集成太多的晶体管 – 基本上是5-6个晶体管才能保存1个比特，即不利于提高集成度，也无法降低成本。但聪明的人们很快就想到了其他方法，使得要存储1个比特，只用一个晶体管和一个电容就能办到。

最早的电容器是18世纪发明的莱顿瓶，通常人们也把它看成人類历史上第一个电池。荷兰的莱顿城的莱顿大学，穆欣布罗克教授发明了一个大瓶子，它的内、外壁贴着一层金属箔，那是电学发展的早期，当时人们已经懂得摩擦产生静电，穆欣布罗克无意中发现，通过摩擦产生的静电可以在这个特殊的瓶子里储存起来，人们将这种瓶子称为莱顿瓶。其实它就是两个分开的金属板，是的，最普通的电容器就是由两块金属板隔着一定距离构成的。另外，同样不用怀疑的是，随便两根电线摆在

一起也是一个电容器；两个人站在一起还是一个电容器；当你在厨房用铁锅炒菜时，你和铁锅之间也构成了一个电容器。

所有电容器都有一个特点，那就是把它接到电源上时，在电源的作用下，一个金属板上的电子会被拉到另一个金属板上，从而，当电源撤走后，这两块金属板会保持着一块电子多而另一块金属板电子少的状态，以致于在它们之间存在电压。如果仅仅从感官上判断，这两块金属板可以储存电，这就是电容器的由来

听起来是一件可怕的事情，因为所有的金属都可以构成电容器，要是它们之间存在电压，那将是很危险的。当然，有时确实很可怕。不过印象中距离很近的导体随处可见，也没有什么危害，这是为什么？

事实上，电容器容量的大小既和两个极板的面积与距离有关，也和它们中间都填充了些什么东西有关。如果极板面积不大，而且中间隔着空气，也没有充过电，或者充电电压很低，那就没有什么危险。但要是你从电视机里拆一个大家伙可以试试用手摸一下它的引脚，相信一定会给你留下深刻的印象，在这种容器里，有着储电效果非常好的电解液

充了电的电容器可以通过其他导电体放电，这相当于一个电池。确实，在有些计算机上，或者一些电子产品里（手机、电子表等）通常用电容来短时间维持电路的工作，当你把手机电池取下来后，日期和时间能维持一会，要是过了很长时间才把电池安上，可能需要重新设定日期和时间了

放电的速度取决于两个极板之间的电阻，可以利用电容器充放电的特点来保存1个二进制比特。充了电的电容器相当于保存了1，而没有充电的则是0，这样，用一只具有开关效能的新型晶体三极管和一只电容就可以存储一个比特，当外部的地址译码器选中这个单元时，三极管打开，电容器可以通过它向数据线放电，或者从数据线上接受充电，这分别相当于读取和写入。历史上第一个采用单只晶体管制造存储器的人是罗伯特·登纳德，他在1968年申请了专利。

电容器在充电后，即使放在那里不用，也会通过隔在两极间的空气或其他介质缓慢放电，这称为泄漏。尽管它有这个讨人嫌的毛病，但用人之道是发挥人的长处，而不是整天盯着人的短处。由于晶体管和电容器可以做得非常微小，这样就能得到密度和容量很大的集成电路存储器，而且成本低，价格便宜。不过让我们感到惊奇的是这样微小的电容器泄漏得更快 – 通常在几毫秒或几十毫秒就泄漏没了。由于这个原因，这种存储器必须以极快的速度定时重写，这称为刷新。也正是因为这个原因，这种存储器也称为动态存储器（Dynamic Random Access Memory, DRAM）。

半导体存储器有一对孪生兄弟，除了RAM还有ROM – Read Only

Memory，意思是只能读的存储器。除了无法把数据写入每个存储单元以外，它的RAM一样，可以通过给出地址而读出任何一个存储单元的内容。

只读存储器最大的优势在于可以一直维持它所保存的内容，即使去掉电源之后也是如此。但如果既能在不需要电源的情况下不丢失数据，又能在需要的时候擦掉重写，可能更符合人们的心意，所以只读存储器一直也在发展变化中，最早的只读存储器是永久不能擦除重写的，它的内容在制造时就已经固化到存储器中了，很显然，如果用户买回存储器后想自己向里面存储一些数据，就无能为力了。所以后来发展出可编程只读存储器，用户可根据自己的需求，对里面的内容重写一次，后来又发展出可擦除可编程只读存储器，可以根据自己的需要随时擦除重写，写完之后照样不会因为断电而丢失数据。这种只读存储器的一个典型产品是闪存，从MP3播放器到存储卡，或者U盘，用的都是这种材料

只读存储器的用处很大，事实上，我们的计算机从来就没能离开过它，例如，可以用它来代替那些复杂的逻辑电路，以实现相同的功能。

我们知道，逻辑电路可以根据不同的输入产生不同的输出，比如全加器，当输入不同的三个比特时，就会在另一端输出一个“和”及一个进位，而译码器也是一个很好的例子。

传统上，所有的逻辑电路都是由与、或、非门构建的，取决于它的功能，逻辑电路可以很简单，只包含有限的几个与、或、非门，也可能很复杂，需要几十个、几百个甚至更多的与、或、非门。

想想看，对于一个特定的逻辑电路来说，每一组输入都会在其的另一端产生你所期望的、设计好的输出。如果把所有不同的输入看成是存储器地址，同时把它们对应的输出固化在存储单元里，不是就可以取代传统的逻辑电路？

这是个好主意，这样一来，一个全加器就可以设计成具有8个存储单元，每个存储单元2位的只读存储器，如图13.2所示

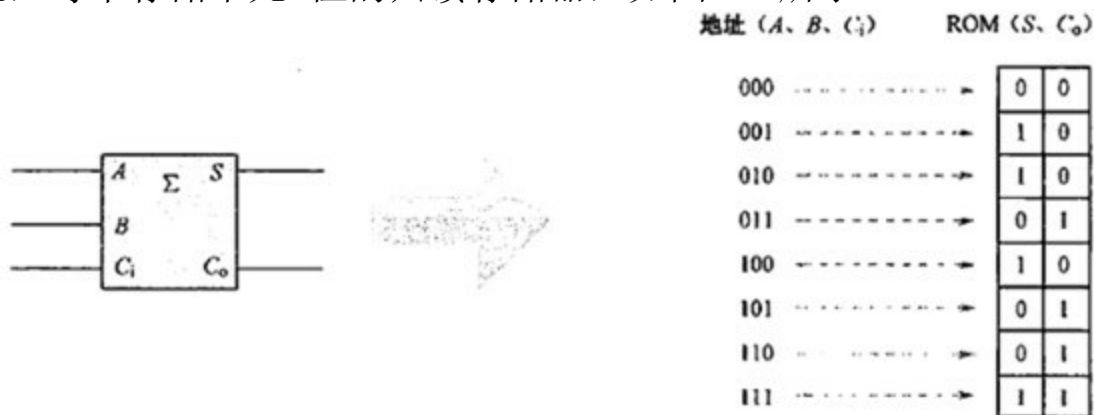


图13.2 采用只读存储器来取代传统的逻辑门电路

用ROM来实现全加器的功能，这并不是一个十分复杂的例子，我

们可以想象到，计算机内部的控制器是非常复杂的，因为它要应付一大堆指令，为它们产生不同的操作序列。最早，控制器全部采用逻辑门搭建而来，后来，在许多计算机上开始舍弃这种方法，转而采用ROM，也就是我们在有些专业书上看到的“微代码ROM”，它比我们这个全加器的例子要复杂成千上万倍，用只读存储器来代替传统的逻辑门电路，最大的好处就在于如果计算机的设计发生了变化，也能很快方便地修改它的功能，而不会有拆掉所有零件然后重新组装的麻烦。如图13.3所示

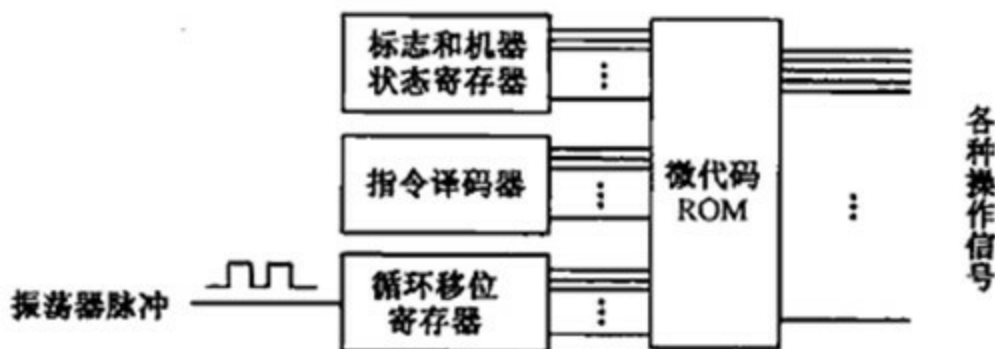


图13.3 现代计算机的控制器结构

传统上，硬件就是制造出来并定了型的逻辑门组合，软件就是驱动这些硬件的指令。显然，在集成电路时代，尤其是在可编程逻辑器件大行其道的现在，这两者之间的界限变得越来越模糊了。通过对ROM编程，可以根据需要改变它的输出，使它的功能发生变化，这就是可重构硬件。

一台计算机可以看成是紧密团结、有效协作的大家庭，它的核心成员包括存储器、运算器、控制器以及一些门电路和触发器。存储器已经被集成化，人们把运算器和控制器也集成到一个单独的芯片里，这样就形成了人们常说的微处理器，这个名称非常恰当地表明了它的体积和具备的功能。微处理器的硅片很小，可能比火柴头还小，但却包含了成千上万、几千万甚至几亿个晶体管，当然，微处理器必须配合其他电路才能发挥作用，所以要给它加个壳，封起来，向外引出一些导线（引脚）

微处理器更多地被称为中央处理器（Central Processing Unit, CPU）。但后者似乎已经被专门用于指计算机上的微处理器。要知道现在手机、电视机、MP3、微波炉、电冰箱和汽车上都用上了微处理器。尽管从表面上根本看不出来，但本质上它们都是一个个的微型计算机，通电后它们都在不停地工作，执行预先存储的程序指令，使你能够保存电话号码、玩手机游戏，或者智能调节冰箱的温度

微处理器或中央处理器都需要存储器及其他辅助设备才能工作并发挥作用，理所当然，它需要向外部提供一直地址线和数据引线，另外还包括一些控制信号线。最起码，它需要有引线从外部获得能量让自己活

跃起来，最后，微处理器没有自己的振荡器，所以它需要从外部引进时钟信息。说来说去，一个现代的微处理器，在封装好之后会有大量的引出线供外部与之相连，图13.4就是一个非常直观的实例，注意那些密密麻麻的小圆点，它们都是引线（引脚）



图13.4 这是一款以前的CPU，中间黑色部分是它的核心，  
周围分布的小圆点是它的引线（脚）

集成电路的发展速度很快，随着技术的进步，集成度越来越高，工作速度越来越快，而价格则越来越便宜。一个针尖上已经可以容纳3000万个45纳米大小的晶体管；此外，现在的处理器上单个晶体管的价格仅仅是1968年晶体管价格的百万分之一。

除了集成度和工作速度外，微处理器的另一个特点是它们都有各自的指令集。比如，对于甲公司生产的微处理器来说，它有一条指令：

1000100111011000

但对于乙公司生产的微处理器来说，这可能根本就不是一条指令，或者这条指令完全是另外一个意思，完成的是另外一种不同的工作。

这意味着，我们国家要生产自己的微处理器，指令集可能是一个需要慎重考虑的因素，除非它不打算运行现有的各种软件

集成电路具有很多优点，但就目前的现实情况来看还不可能完全代替采用独立零件的电路。一是有些东西，比如大的线圈还没有办法集成；二是集成电路因为其微小的缘故，不能承受大的电流和高电压，这就限制了它只能出现在像电子表、手机和其他一些更适合随身携带的设备中，或者大型设备中功率较小的那一部分电路里。如果电流过大，集成电路就会烧毁，就像熔断器扬起的作用那样，世上再也找不出像集成电路这样构造复杂、制作精良的熔断器了

## 13.3 流水线和调整缓存技术

在对微处理器有了一个大致的了解之后，我们可能需要再次回过头来，审视一下它的存储器之间协同工作的情况

微处理器的速度很快，而且随着时间的推移和技术的进步，它会越来越快，尽管振荡器（时钟）的频率不能完全代表处理器的速度和性能，但还是具有参考意义的。几十年前，处理器还在几兆赫兹（MHz）的频率下运行，但现在这个数值已经提高到几吉赫兹（GHz），增加了1000倍，为了直观，我们通常更喜欢用每秒钟可执行的指令数来衡量处理器的速度，这个数值在现实中的处理器上是几百万条指令每秒到几亿条指令每秒

处理器当然有许多事情要做，但这些事情大都需要一系列步骤才能完成——从存储器取指令、译码、读/写操作数、移位、加减乘除，以及其他任何需要的操作。理想情况下，当前一个步骤完成时，后一个步骤应该紧随其后，中间不应该存在时间上的延迟。要是这样的话，我们手头上的绝大多数工作都应该能在瞬间完成，但事实上却感觉不到这种情况的存在，处理器当然非常快，但它不是在孤立地工作，需要一大堆外围设备的配合，为它提供数据，遗憾的是这些东西相比处理器的处理速度都很慢。比如，每次用U盘复制文件的时候，总是会看到一个进度条，这不能怪处理器速度慢，而是U盘不够快

U盘这种东西以后会讨论，不是我们现在的主要话题，我们知道，离处理器最近的是存储器，如果说处理器是加工厂的话，那么存储器就是原料和成品仓库。在后面的章节中我们还会看到其他类型的存储器，但在历史上，它们在离处理器很远的机箱外，为了加以区别，和中央处理器最近的存储器通常称为主存储器或内存储器，简称内存

论访问速度，由触发器构成的内存（SRAM）最快，一般为几纳秒。与之相比，动态存储器（DRAM）差些，访问速度可能是几十纳秒，很大一部分原因在于它需要频繁地刷新，在此期间无法接待处理器的造访。

但与SRAM相比，DRAM最大的优势在于它的高密度和低成本，使得我们可以花很少的钱就能买到一个大容量的内存，对于个人应用来说，这是一个很好的折衷方案，即不会慢到无法忍受，同时又省了钱，问题是，处理器就遭罪了，理想情况下需要7个时钟周期的指令，可能实际需要50个时钟周期才能完成，多出的这些时钟周期，完全是为了等待内存而临时插入的（图13.5）



图13.5 由于存储器的速度很慢，CPU经常处于等待状态

处理器是比较昂贵的资源（当然，整台计算机都是），昂贵的资源应该保持忙碌才行，这样才对得起为它付出的时间、空间、金钱和电力成本。在这种情况下，为了让中央处理器满负荷地工作，流水线操作是必然的选择。

从某种意义上来说，流水线是计算机里的一个必要的恶魔，有人这样提到 – 它存在的理由是要在中央处理器必须完成的工作和所需时间之间找一个平衡点。让我们来看一个例子

我们已经知道，电流的速度是每秒30万千米，计算机的速度也很快，通常工作在纳秒（ns）甚至皮秒（ps），1秒=1000000000纳秒或1000000000000皮秒。所以在1纳秒、1皮秒的时间内，电流只能分别向前传播30厘米和0.3毫米

一旦在大脑中有了这样的概念，再来看看，假定有一个字节的数据X需要用两个步骤加工成Z（图13.6）

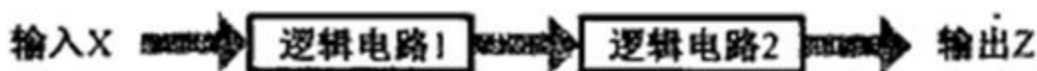


图13.6 数据加工的总时间是所有单元加工时间的总和

在这里，逻辑电路1和逻辑电路2分别用于完成那两个加工步骤，再假设，数据通过这两个逻辑电路需要相同的时间都是50皮秒，那么从X到Z的整个传输延迟就是100皮秒（两个逻辑电路之外的传输延迟可忽略不计）

而且在数据加工期间，X必须一直保持，起到100皮秒后Z出现在右边的输出端，然后X才允许换上新的数据并开始下一轮加工，只有这样，才能确保输出Z是稳定和正确的

这意味着，每个X的加工时间都是100皮秒，同时也意味着，每隔100皮秒，我们才能从右边看到一个新的输出Z

为了改善整个电路的数据加工速度，流水线可能是一个不错的选择。因为X需要经过两次加工，所以这可分为两级，每一级的加工结果都用寄存器保存，寄存器的作用是隔离两个加工级别，并使下一级的加工稳定可靠（图13.7）



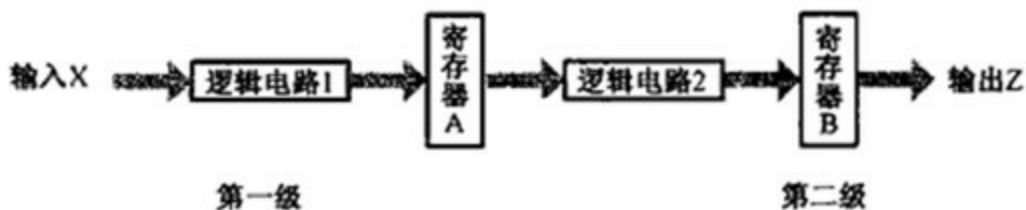


图13.7 同样的设备，采用流水线可缩短加工时间

一开始，X被逻辑电路1加工，50皮秒的延迟后，数据到达寄存器A并被锁存，寄存器的运作需要一点点时间，为了便于说明，可以忽略不计

紧接着，从第50皮秒开始，第一级的加工结果由寄存器A保持，并被逻辑电路2加工，由于寄存器A的存在，输入X不再需要保持，第一级实际上是空闲的，它完全可以在第二级启动的同时加工新的数据

同理，当第一个X的加工结果出现在寄存器B并被可靠地锁存时，第二个X正在被第二级加工，第三个X正在被第一级加工。尽管每个X的加工时间不变，还是100皮秒，但我们却可以每隔50皮秒就能在寄存器B得到一个新的Z，而不是以前的100皮秒，这正是流水线的妙处

为了在处理器中使用流水线技术，可以将整个指令的执行过程分为三级：取指令、译码、执行，让它们重叠执行（图13.8）

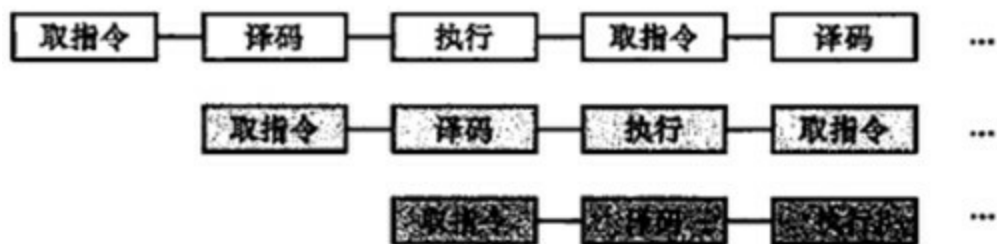


图13.8 超标量体系结构

使用流水线技术，现代的计算机可以改进其设计和结构，允许在一个时钟周期内运行多条指令，这称为超标量体系结构。尽管看起来很了不起，但是流水线技术并不如我们想象的那么完美，有很多潜在因素会影响它的效率。比如对于一条跳转指令，当它开始执行时，后面的两条指令已经进入流水线，在这种情况下，处理器只能清空流水线，从将要跳转到的目标地址那里重新读取指令

解决这个问题是为处理器增加预测功能，通过为处理器增加额外的电路，来预测将要发生的跳转。分支预测不会百分之百成功，但总比猝不及防要好得多。充其量是要清空流水线，使处理速度变慢，但不会比这更坏

除了流水线，另一个被用来平衡处理器和内存速度的手段是使用高速缓存技术，字面上的意思是速度很快的缓冲存储器。类似于蓄水池，这种技术基于计算机运行的一个特点- 局部性，通俗地说，局部性的意

思是程序在被执行的过程中常常会访问最近访问过的数据，或者该位置附近的数据。

SRAM的优点是速度快，但制造成本很高，通常不作为内存使用。不过，好的东西不能拥有全部，来一点点应该还是可以的。基于局部性的原理，可以在处理器和内存之间放置一小块SRAM，当处理器从一个新的内存地址开始执行，将那一整片的东西都搬到这块SRAM中，这样一来，如果下次要访问的内容正好在SRAM中，就不用再到内存中去取，从而节省了时间（访问内存比访问SRAM需要更长的时间）。传统上，这一小块SRAM就是高速缓存，也就是我们经常在技术文章中看到的单词cache

在实际应用时，高速缓存有多种组织方案，但从直观上来说，它好像两张表格，第一个表格存放的是从内存取来的一块块数据；第二个表示则存放每一个数据所在的内存地址，如图13.9所示

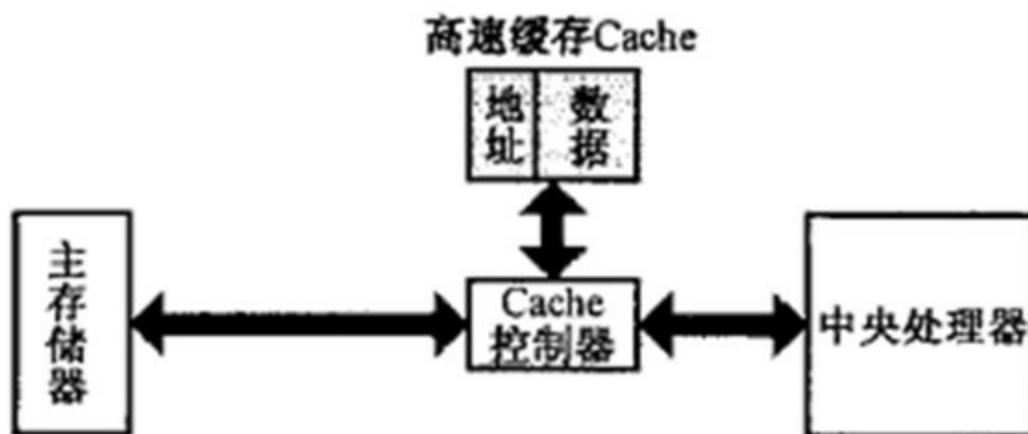


图13.9 高速缓存示意图

高速缓存有一个控制器，即图中所示的cache控制器，平时，中央处理器送出地址，要从内存中取数据时，cache控制器把该地址同高速缓存中的地址进行比较，以查明该数据是否在高速缓存中。由于大规模集成电路技术的发展，地址比较花不了多长时间，非常短暂

如果要取的数据正好在高速缓存中，那么，很好，这称为“命中”，处理器可以直接拿到数据。处理器从高速缓存中取得数据的时间通常很短，称为“命中时间”。相反，如果数据不在高速缓存中，称为“不中”。

高速缓存不中是非常糟糕的，是最坏的情况。在这种情况下，处理器需要重新装载高速缓存（而不是硬着头皮却访问内存）。装载高速缓存也需要时间，即“不中惩罚”，这意味着执行一条指令需要更多的时间）

高速缓存是否能大幅度提升计算机的速度和性能，取决于命中的概率，也就是命中率，实际上，命中率受很多因素的影响，包括高速缓存

的设计和软件的编写技巧，前者通常是硬件设计公司的机密，后者需要软件工程师的智慧

## 13.4 掌上游戏机和手机就是计算机

最早的电子计算机出现在电子管时代，它们的何种都很大 – 这是没有办法的事，晶体管和集成电路的出现为时尚早。不过“大型机”的称呼倒是保留至今，尽管已经没有什么面积和体积的内涵

在那个时代，计算机在人们心中就像是卫星、火箭……不是为普通人发明的玩意，只有那些称得上是“问题”的问题上才配在这样昂贵的大家伙上计算一下。用它来上网？那个时候还没有互联网。玩游戏？那个时候也没有什么有趣的电子游戏，上机时间非常宝贵，分分秒秒都要付费，在这样的计算机上玩一个哪怕是现在看起来很简单游戏都是一种罪过。所以，几乎没有人认为大众会需要这种东西，这是很自然的。

昂贵的，普通人用不着的东西不需要多制造，但政府和大型企业双需要它，在这种情况下，大型机的目标就是计算能力更强、速度更快，甚至一台机器应该为一大堆用户服务，同时干好多工作，这就是所谓的分时、并行、多用户和多任务处理。为了这个目标，大家干得很起劲。很多我们现在依然在使用的技术，包括一些看上去很新、很时髦的技术，其实在那个时候已经有了，只是缺乏大规模普及的条件。到了20世纪六、七十年代，集成电路发明了

在实际拥有一部智能手机之前，有谁会想到智能手机的好处？又有谁会想到它可以打电话、上网、听音乐、看视频、玩游戏？几乎没有。甚至也没有人想到它对我们如此重要。个人计算机也是如此，集成电路的发明使一部分人看到了电子计算机小型化的前景，但不十分清晰，在那个时代，认为每个人都会拥有一台计算机需要的不仅仅是超前的眼光，更要有直面被人骂成是白痴的勇气。这时的关键是：虽然集成电路可以缩小计算机的体积，但为什么每个人都需要这么一样东西？

在这种情况下，难怪小型机的创始人奥尔森在1979年十分肯定地说“没有理由让某个人在家中配备一台计算机。”奥尔林是美国IBM公司的总裁，还发明了小型机，本来就很有名，这下名气更大了。

在国内，个人计算机（Personal Computer, PC）以前叫微型计算机，简称微型机、微机或微电脑。显然，这已经成为历史名词，个人计算机兴起于20世纪70年代，那时，世界上第一个微处理器已经问世，但功能很弱，所以这些个人计算机的鼻祖怎么看都像是玩具

正如大家已经看到的那样，技术总是在不断地向前发展，随着微处理器的速度越来越快，功能越来越强，存储器的容量也更大，速度也更快，以往只在大型机上使用的先进技术，比如流水线和高速缓存，现在也用在了个人计算机上以提高性能，以前只在大型机和小型机上解决的

任务，现在也可以在个人计算机上完成。总之，它们之间的差距正在慢慢缩小

越来越小型化、功能越来越强大，这就是未来的发展方向，智能手机的问世，可以放在手上的电视机、音乐播放器、视频播放器和游戏机问世。尽管发明这些东西所需要的理论和技术几十年前就已经完备，但使它们小型化、微型化，也只有现在的技术条件才能做到。不管任何东西，只要稍微有一点智能 – 从录音笔、微波炉调温装置、电冰箱、电子表、智能手机、游戏机，都需要用到微处理器和一小块内存存储器，也需要一些编排精巧的指令，你能说，那些大家伙叫计算机，而这些东西就不叫计算机吗？

## 第14章 核心与外部设备间的接口

在对计算机核心部分的工作原理有了相当的认识后，应当意识到一台计算机仅仅只有中央处理器和内存是不够的，它当然能够运转，但没有什么大用处。

想想看，按老式的说法，你得用开关把程序指令一条一条写入存储器，这显得既笨拙又单调，更不要说用开关来编辑文稿，会多么恐怖！而且，还无法直观地看到处理器产生的结果。

上面说的是两个最基本的总是，也就是输入和输出。为了取代开关，我们现在用上了键盘；为了知道计算机忙碌的结果，我们还发明了显示器，这是两样最基本的输入、输出设备，马上要详细地介绍它们的神通，尽管我们在前面从来没有提到过这两样东西，但用过计算机的人都不会陌生。

没有输入和输出设备我们将无法向存储器写入指令，也看不到执行的结果，但这还不是最重要的。尽管发明计算机的原因正如它的名字那样，是为了解决计算总是，但这也正是它在其他方向显得没有大用的根本原因。不错，它能越来越快，越来越好地完成人类几千年的夙愿，解决一个一个计算难题，但人们逐渐觉得它还应该能做更多的事情，比如娱乐、文本处理、控制车间里的机床……，不过想要达到这些目的，还需要更多的，各种各样的能够和中央处理器、内存这些核心通信的输入和输出设备

当然，这不是最近才冒出来的想法，20世纪70年代，美国卡耐基-梅隆大学计算机系的两名研究生就曾经搞过一项古怪的发明，这两人酷爱喝可乐，还非得是冰镇的，麻烦在于可乐机在三楼，离他们很远，有时当他们去可乐时，不是已经没有了，就是还不够冰，这让他们觉得很麻烦，于是，他们想了一个主意，因为这台可乐机有6个冰柜，每当有可乐被送出的时候，灯就会闪烁，而一旦冰柜空了，灯就会一直亮着。利用这一点，他们用电线把灯连到一台计算机上，并编写了相应的程序指令，这样就可以随时查询哪个冰柜里的可乐最凉。很显然，这就是一个输入、输出的例子，这项发明一直被用了十多年，期间还经过不断改进，一直到互联网逐渐开始兴起，借助这门新兴的技术，另外一些疯狂的家伙又把这套东西搬到互联网上，即使是千万里之外也可以查询可乐机的状态

## 14.1 计算机同外部的接口

专业地说，输入、输出设备称为I/O设备，因为这是输入、输出分别对应着英语单词Input和Output，它们位于计算机的核心之外，所以也称外部设备。除了我们熟悉的键盘、鼠标、显示器，数码相机、智能手机、MP3播放器也是外部设备。外部设备是庞杂的，几乎包括了任何东西，取决于是否希望它们和计算机核心产生联系。计算机因为外部设备而变得用途更广，计算机核心部分要从外部设备那里获取自己需要的数据，知道外面的情况，最起码知道自己是不是正常，而外部设备呢，也应当理解计算机内部发来的控制数据，采取适当的措施来控制自己的行为

我想这里需要一个实实在在的例子，比如一个全自动的温度控制系统，通常，设计这样一套系统的目的是用于炼钢、孵化器、集成电路制造等

全自动温度控制系统的用处是让温度保持在一定的范围内，当温度低于某个值时就接通加热器；如果温度过高，及时停止加热。在这个过程中，计算机所要做的就是不停地监测温度，发现温度低了，通常加热系统加热，或者在温度过高时通知加热系统降温

那么如何从现场取得温度呢？要知道，中央处理器和存储器不认识温度计，无法感知冷热变化，所以我们只能将温度转换成相应的电压和电流，这个比较容易，通常采用一种叫热敏电阻的东西，和铅笔芯这样具有电阻的物质一样，热敏电阻可以降低电压、减小电流，但奇特的是它的电阻会随着温度的变化而增大或减小，从而使得电压和电流也跟着改变。

这还只是第一步，因为温度是连续变化的，因此得到的电压和电流也是连续变化的，这称为模拟信号。模拟信号只能用大小、高低和强弱来衡量，计算机不接受它们，计算机只和二进制数打交道，也就是以开关形态出现的脉冲，所以，要想让中央处理器认得它们，还必须将模拟信号进一步转换成数字信号，这一转换过程如图14.1所示



图14.1 从温度到数字的转换过程

从模拟信号转变到数字信号 – 也就是现在非常时髦的“数字化”，需要在一个小电路里进行，无非是一堆电阻以及一些逻辑门。它所做的工作很简单，就是每隔一会儿“观察”一下模拟信号的电压（不管在什么地方，要定时做某件事必须用到振荡器，用它的脉冲来控制），然后把电压的高低大小表示为一个二进制数。比

如，没有电压表示为00000000；电压小于1V表示为00000001，电压在1-2V之间表示为00000010.....

看起来挺简单，但这里面还是有一些讲究的，首先，所有的计量器具都不是全能的，都有各自的量程，不可能用一杆秤就能称量1毫克和1000万吨。一样的道理，根据适用的场合，模拟和数字转换电路有不同的转换精度，同样是模拟电压在1V和10V内波动用10个二进制数来区分它们和用100个二进制数来区分是不一样的，后者需要更灵敏、更复杂的转换电路。

除了精度外，还有一个问题是速度，而且速度往往会影响精度。在有些场合，数字化后还要还原，即把二进制数重新变回模拟信号，一个典型的例子就是听MP3音乐，MP3播放器里有容量很大的闪存，要听歌曲、相声什么的，需要事先将歌曲、相声保存在闪存里，不过在此之前，人们必须将这些歌曲、相声从声音转变成二进制数，这就是前面所说的模拟-数字转换。

当对着话筒说话或播放音乐时，将产生强弱随时间快速变化的电流。这时，需要用模拟-数字转换电路数字化，也就是不停地观察模拟信号的幅度，并得到一个二进制数，称为“采样”（图14.2）

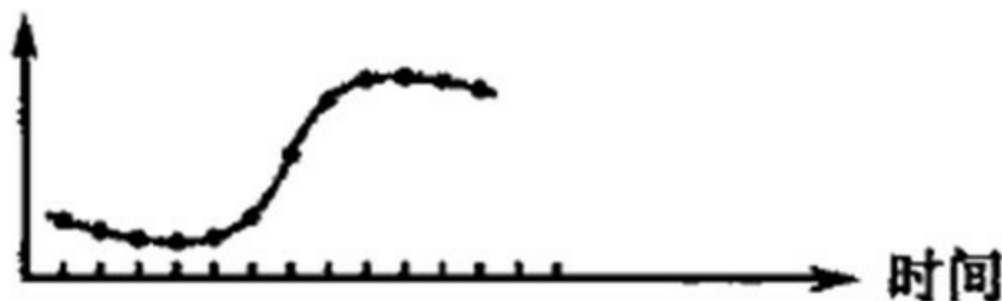


图14.2 数字化的过程就是用数字来代表电压的幅度

“采样”就是“采集样本”，这实际上是借用了统计学里面的概念。声音是快速变化的，如果一秒钟只观察2次或10次，得到2个或10个二进制数，就会漏掉声音中绝大多数美妙动听的部分，将来还原之后只能听到一些古怪的声响。为了保证数字化的质量，需要尽可能地提高每秒钟采样的次数，这称为采样频率

就现有的标准而言，高品质音频的采样频率是44.1kHz，这是什么意思呢？因为模拟-数字转换电路没有智能，所以它需要一个报振荡器来驱动、定时。振荡器的频率是44.1kHz，所以它每秒可以要求采样44100次，生成44100个二进制数。如果是立体声，就是44100\*2=88200个。通常，每个二进制数是16比特，也就是两个字节，换算一下，一个立体声音频每秒钟将产生176400字节的数据，接着换算，一个3分钟的歌曲将产生30M字母的数据

即使是在数据存储技术突飞猛进发展的今天，像这样生成数据的方法



法也会对如何保存它们形成一个挑战，为了在光盘这样的介质上存放电影和音频，1988年，国际上成立了运动图像专家组（Motion Picture Experts Group, MPEG），专门研究活动的视频压缩方法，以减少它们的数据量

MPEG不仅定义如何压缩视频，还定义压缩音频的标准，有一个音频压缩级别，分别对应于不同的压缩比率，其中第三个级别是最常用的，在这个级别下音质和压缩比率之间的搭配最优，称为MPEG Layer3，或MP3，这就是MP3的由来。视频和音频的压缩是有损失的，但应该有个限度，不能超出眼睛和耳朵的容忍范围

MP3以及其他各种随身听装置虽然能够随时随地欣赏音乐歌曲，但长期使用它们却会造成伤害。如果使用耳机音量太大、听得时间太长，听力会受到操作，而且不容易恢复，听力下降、耳朵疼痛

模拟到数字或者反过来数字到模拟的转换过程处理器无法胜任，这不是它擅长的业务，它只知道把二进制数搬过来移过去，然后运算。所以我们需要单独的、专门的电路来完成这项工作，但无论如何，温度会变成电压和电流，然后再变成二进制数，来到计算机，这是毫无疑问的。

在计算机这边，需要专门为温度控制编写一套程序指令，想想看，中央处理器唯一擅长的就是执行命令，外边来的数据代表什么意义、怎么处理，只有工程师们知道，他们必须把自己头脑中的思想和意图，也就是控制温度的方法变成指令，计算机才能应付这一切，该程序指令应当按下面的步骤来执行：

- （1）从外边把温度数据移动到RA寄存器中
- （2）将RA中的值和正常的温度值（一个已经确定的立即数）比较
- （3）若正常，跳转到第（1）步（继续监视）
- （4）若温度低了，向外部设备发送表示打开加热器的二进制代码，然后跳转到第（1）步
- （5）若温度高了，向外部设备发送表示关闭加热器的二进制代码，然后跳转到第（1）步

现代的计算机有那么丰富的指令，可以轻松地执行上面的步骤。麻烦在于，截止到现在，中央处理器只和内存打交道，从内存中取一个数，或者把一个数写入内存，这都可以，但它怎样才能得到从外部设备来的数据呢？用开关手工操作？太繁琐了！

为了在外部设备和计算机核心之间传送数据，需要在这两者之间连线，并构造逻辑电路。在逻辑电路里，有一些寄存器，通常称为I/O端口，或直接称为端口。当中央处理器有话要对外部设备说时，就把它放在端口上，由外部设备取走；当外部设备有话要对中央处理器说时，也

照此办理。从形式上看，端口类似于传达室。

端口是内外交流的窗口。但既然是窗口，一个肯定不够，最好是每个外部设备都有好几个。同理，在计算机上可能同时存在着好几种不同的输入、输出设备，如键盘、鼠标、显示器、打印机、音箱.....不可能轮流使用它们，用键盘的时候把键盘插上，用鼠标的时候拨下键盘插上鼠标，所以，尽可能为每种外部设备都配备一些端口是非常有必要的。至于每个端口怎么用，这无关计算机的制造者，取决于：

(1) 外部设备打算怎样使用这些端口，端口上的数据都有什么用；

(2) 正在中央处理器上运行的软件，它们如何解释从端口上读来的数据

可以想象，如果没有与外部设备对应的软件，那台设备将会一直在那里发呆，即使它将数据放在端口上，也没有谁会来把它取走。相反，如果只有软件，而没有与之对应的外部设备，那这个软件无论发送什么命令，也得不到任何回答。这就是为什么我们有时在屏幕上看到一个对话窗口，说“设备没有响应”，或“设备不存在”

计算机内的所有端口都应当像内存地址一样顺序编号，以方便读/写。但，不管有多少端口，怎么编号，现在我们必须回答刚刚提出的问题：中央处理器怎样才能访问得到端口呢？

为了能访问端口，从端口读/写数据，第一种办法是合理地布线，并重新设计地址译码器，将端口并入内存的地址空间，将其看成内存的一部分来读/写，比如，如果要访问的内存地址位于00000000-11110000之间，那么这是真正的内存单元，如果是11110011它就是一个I/O端口

这是一个好办法，但只在某些类型的计算机上 useful。对于个人计算机来说，更普遍的做法是为中央处理器增加新的指令，专门用于访问I/O端口，毕竟它们和存储单元不是一回事，比如：

IN RA, 61

这条指令的意思是从61号端口把数据读入RA寄存器，再比如：

OUT 62, RA

表示把RA寄存器中的数写入62号端口

在这种方案中，处理器发出的地址和数据既送到内存，也送到每一个I/O端口。但中央处理器有专门的控制线和逻辑门来通知由谁来接收这件东西 – 是内存单元还是端口寄存器

从中央处理器的角度来看，外部设备位于遥远的地方，它能看得到的，只是端口 – 那些寄存器。它从端口接收数据，做一些处理工作，或者将数据交给端口，于是这有一些数据在这两者之间穿梭，如图14.3所示



图14.3 端口的位置（处理器的视角）

外部设备确实很“遥远”，毕竟它们通常不是计算机的必要组成部分。但端口到底在哪里呢？有必要打开你家的那台个人计算机的外壳，来简单地看一下它的内部

## 14.2 I/O接口

打开计算机的主机箱，主机的核心是一块电路板，通常称为主板或母板，这里是计算机内部所有零件的大本营。在主板上，你会看到中央处理器，上面固定着散热风扇。现代的中央处理器都在很高的振荡频率下工作，发热量很大

除了中央处理器，还会看到内存 – 通常称为内存条，因为它们是窄长条，插在主板的插槽上。除此之外，主板上的其他零件中，有些时服务中央处理器和内存的，比如电容、电阻、振荡器、各种控制芯片等等；而大部分的东西是为连接外部设备而存在的。

哪些东西是外部设备？很多很多，一只话筒，当你想把它连接到计算机录音时，它就是外部设备。但问题是外部设备应该如何连接到计算机 – 接头。

接头只是一个方面，另一个更麻烦的问题在于接头里的信号，经过话筒插头的是快速变化的声音电流；视频采集设备经过插头信号的是图像的色彩信号和同步信号（决定图像如何显示在屏幕上）；打印设备的插头经过的信号是字符。在这些信号里，有些是模拟的（话筒的电流），有些是数字的（送给打印机的字符），但计算机只能处理二进制

保留已有的接头规格、模拟-数字信号的转换，这是两个最基本的问题。当然还有其他一些问题，比如信号的编码和解码。要想用计算机来录像，那么不但要考虑接头和模拟-数据转换的问题，还必须做一些传统上只有电视机才能做的工作，那就是将视频信号中的各种要素分离出来，很为难吧？没办法，计算机只按它自己的方式行事

办法还是有的，不知道你发现了没有，在现代计算机的主板上，都会一些长长的插槽，这些插槽统称为扩展槽。“槽”的意思好理解，如果你此刻正在观察主板，你会发现，它们确实是一些槽，至于“扩展”它的意思是扩展计算机的功能，使之能做更多的事情。

插在扩展槽上的是一些电路板，它们是为了解决前面提到的问题而存在的，通常称为接口卡，比如，用来建制或插入声音的电路板，称为“声卡”，用来录制或播放视频的电路板，称为“视频采集卡”，用来连接到网线的电路板称为“网卡”。我们现在所用的计算机都有显示器，为了把内存中的二进制数据变成字符或图像，也需要一块电路板，称为“显卡”，不管是什么卡，它们都只是一块电路板，如图14.4所示

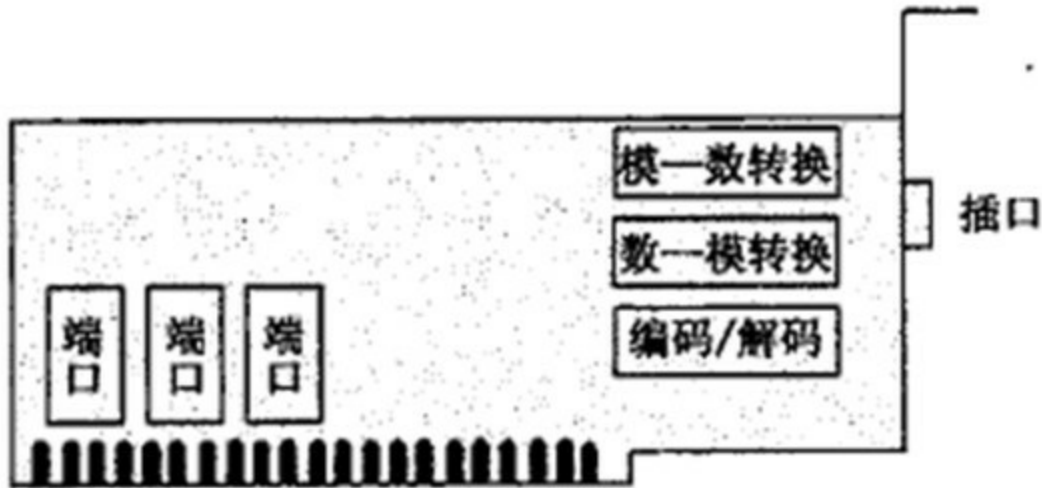


图14.4 计算机接口卡的外观和基本组成

首先，除了外形外，根据外部设备不同，接口卡的电路结构也不一样。毕竟，它是为连接到某种类型的设备而定制的

其次，每种接口卡都有能到外部的接头，或者说插口，它们的形状和构造取决于外部设备是什么。如果要连接话筒、功放或扬声器，那么插口就是小圆孔，可以用来插入一个标准的立体声插头；图14.5就显示了几种常见的插头类型，其中，标有LAN的，是网卡的插口，标有HDMI的，是高清多媒体插口



图14.5 几种常见的插口，不包括DIP SW

在形形色色的外部设备中，有很多是模拟的，比如话筒、温度探测器、扬声器、老式录像机等等。如果它们想把自己的信号交给计算机，则必须先进行模拟-数字转换，即图中的“模-数转换”；反之，如果计算机有话要对外部设备说，则要做相反的工作，即“数-模转换”

除此之外，有些外部设备的信号很复杂，包含多种成分，比如视频信号，这样，一个录制或播放视频的信息的接口卡可能需要包含和电视机相似的电路，除了不需要显像管、调谐器（选台用的）和音频放大器，比如，在这里就是图中的编码/解码电路，它可能是一个或多个集成电路芯片

一旦模拟信号被转换成数字信号，下一步的工作就是将它放在端口上，以便被中央处理器拿到。但，端口原则上并不是主板的组成部分，所以它只可能位于接口卡上，因为只有接口卡才知道它需要几个端口，以及这些端口的用途

如图14.4所示，在每一个接口卡的边缘，有一排像钢琴按键一样的

东西，它们是接口卡与主机连接的导线，当接口卡插到主板的扩展槽里时，这些脚就会与主板上的电路接通

从某种意义上说，接口卡是典型的城乡结合部，是计算机核心与外部世界的中转站和缓冲地带

因为不知道你买回计算机后要接哪些设备，所以理论上，待售的计算机只预留了扩展槽，而没有接口卡，买回计算机后，根据自己的需求添加接口卡

不过，计算机制造者们发现，有些外部设备是几乎每台计算机都会用到的，比如硬盘、键盘、鼠标、显示器等。为了方便自己，也方便大家，很快对于这些常用设备来说，接口卡不再是可选的了，而是在制造一台计算机时，就被永久地焊（集成）在了主板上

从这个意义上来说，用于将外部设备和计算机核心关联起来的那部分电路最好叫I/O接口，而不是一概称为“接口卡”，因为它可能并不真的是一块插在扩展槽上的卡，可能听过串行口和并行口，及USB接口，它们的I/O接口电路都位于主板上，这称为“集成”。

## 14.3 中断和直接存储器访问

计算机的核心与I/O接口相连，I/O接口又与外部设备相连，当这一切准备好之后，剩下的只有一件事情：开始干活吧！

为了和外部设备打交道，计算机内部必须运行程序指令，这是显而易见的，因为中央处理器没有生命，只有程序员知道该如何使用这些设备，也只有他们知道该通过哪些端口，并在这些端口上放些什么才能使设备工作

举一个例子，为了让计算机顺利地晋升为一台音响设备，需要一块声卡，世界上第一块声卡是由新加坡创新科技公司的沈望博于1981年发明的，当时叫做“Sound Blaster”，翻译过来是“声霸”卡。

声卡是一个典型的I/O接口，在外面它与麦克风和扬声器通信，最后这两样都是模拟设备，声卡的工作是将麦克风的音频电流数字化，或者将连续不断的数字模拟化，并驱动扬声器

对于程序员来说，声卡使用的端口号以及每个端口的用处，他们是清楚的，所以他们会编写使用声卡的程序，指示声卡做下面的工作：设置采样频率、开始录音、停止录音、传送声音数据、播放、停止播放、增加或减小音量、设置回声，等等，这些都是通过端口发给声卡的命令

为了执行中央处理器的命令，落实中央处理器的政策，声卡可能需要智能一些，这样，仅仅把一块做成图14.4那样就太简单了，实际上，现代的I/O接口都有自己微处理器，可以执行与设备相关的程序指令

如图14.6所示

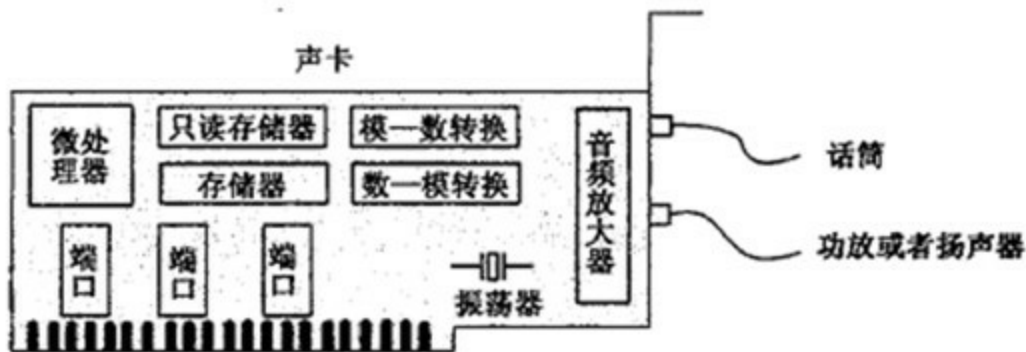


图14.6 声卡的基本组成

声卡有自己的微处理器和振荡器，需要执行自己的指令，而这些指令都已固化在只读存储器中。一旦它通过端口得知计算机让它录音，它就打开话筒的输入线路。因为话筒的输入信号通常十分微弱，所以要先进行放大，然后进行模拟到数字的采样和转换。通常情况下，采样的速度很快，可能来不及通过端口发送出去，所以要先保存到它自己的存储器

里。很显然，这一小块存储器就相当于一个蓄水池。

看起来一切皆好，但这里面有两个问题：首先，当声卡开始录音的时候，它会把代表声音的二进制数据放到约定好的端口上（假定中央处理器和I/O接口通过端口交换数据的速度足够快）。端口只有一个D触发器，任何时候都可以从中读到数据，尽管该数据并不是我们有意写进去的。

这意味着，如果你在97号端口写了数据，还应该在98号端口写另一个数据——这是个标志，如果中央处理器读98号端口，看到了这个标志（通常这是约定好的），就知道97号端口上的数据是有意放置的，注意，这里的97、98号端口只是一个例子，并不是声卡所使用的实际端口号

为了取得这些数据，中央处理器需要运行程序员编写的指令，不停地监视97号端口，起到发现那个特殊的二进制数字。这样，它从98号端口取得数据，并清除97号端口中的标志，然后对取得的数据进行处理。处理完成后，它会中转到前面，接着监视端口的状态，直到发现最新的数据出现。

这意味着，一旦开始录音，或者播放一首歌曲，除非将这些事情做完，否则的话，在这个过程中，我们将无法写报告，或者上网看新闻，因为中央处理器无法分身。

为了充分发挥中央处理器的运算能力，计算机应该能够同时干好多事情，如果计算机足够快，这些事情都能顺利地进行。工程师发明了中断。

中断的意思是在做一件事情的时候临时打个岔，中途去做另一件事情，然后再回来，这好比拍了一下中央处理器的肩膀，告诉它这里有一件事需要它过来处理一下，在有些计算机原理书籍上，把中断看成你正在吃饭，突然电话铃响了，于是你放下碗筷去接电话，然后再坐下来接着吃饭。这是一个很好的比喻，非常恰当

为了获得中断的益处，需要在I/O接口和中央处理器之间架设专门的线路，用来传送中断信号。而且，每种外部设备都分配了各自不同的中断号，一旦有了中断，计算机的内存里可以放置多个不同的程序，而不是像以前那样每次只能有一个。

和以前一样，中央处理器每次依然只能执行一个程序，但是，当中断发生的时候（比如它来声卡），中央处理器就中断当前程序的执行，把下一条指令的地址，连同各个寄存器的内容都临时保存起来，这叫做“保护现场”。然后，它响应中断，跳到另一个程序那里接着执行。

注意，中央处理器只知道闷头干活，为了使它不至于迷路，每个利用中断工作的程序，在它的最后都应当是一条中断返回指令。



这样，当声卡有数据需要传送的时候，它就会扯扯绳子，给中央处理器发送一个中断信号，告诉中央处理器有新的数据产生了！于是中央处理器放下手头的工作，来接收声卡的数据并进行处理，完了呢，再接着继续以前的工作。

中断可能来自任何地方，对于现代的计算机来说，中断是无时不有的，在你一眨眼的时间，差不多就发生了无数的中断，比如键盘上的某个按键被按下，鼠标动了一下，等等。别的不说，光是计算机内部的那个钟表，它每隔55毫秒左右就要在中央处理器的肩膀上拍一下，据我所知，面对无何止的搅扰，处理器的脾气可能是最好的。

中断是最早在计算机中采用的技术之一

对于声卡来说，不管是录音还是播放，数据都要通过中央处理器内部的寄存器中转，从内存到寄存器，再到外部端口或者反过来。尽管中央处理器的速度很快，但当它不得不面对行动迟缓的外部设备时，就慢下来了。

在这种情况下，I/O接口应该拥有自己的本地存储器，既然来不及把数据传送到计算机核心那里，就先放在自己旁边，到一定时候，它再启动一个存储器到存储器之间高速、直接传送，称为直接存储器存取（访问）

直接存储器存取简称DMA（Direct Memory Access），是现代计算机另一个工作机制。当DMA传送开始的时候，所有无关的部件都被禁止出声，让出地址总线 and 数据总线供外部设备与内存交换数据。

在传统的端口工作方式下，程序员编写的指令控制着一切，包括从I/O接口那里取来的数据放在内存的什么地方（地址），即使是采用DMA传送，源数据在哪里，数据传送到什么地方，什么时候开始传送，也必须是在程序的控制下进行，唯一不同的只是速度

## 14.4 键盘

对于一台完整的计算机来说，键盘是必不可少的。从外观上来看，键盘很简单。

键盘的祖先是打字机，最早的打字机大约发明于19世纪初，从那以后，它不断被改进以方便使用。

如果深入了解键盘，会发现它其实本身就是一台计算机。

把键盘看成计算机并不夸张，键盘有自己的微处理器，它有一些引脚连在各个按键下面的开关上（在表面上看不到），如图14.7所示

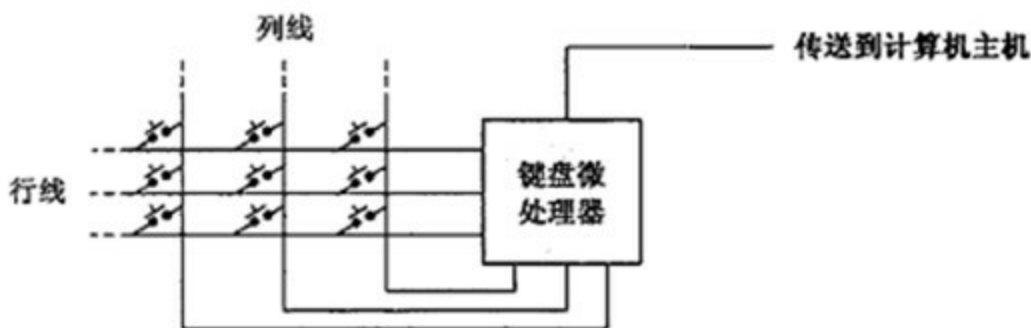


图14.7 键盘电路

个人计算机的键盘微处理器有些特殊，它不但具备处理器的功能，还在内部集成了动态存储器和只读存储器，可以执行自己的指令。按键开关按行、列的方式组织，键盘加电后，键盘微处理器开始扫描行线和列线，以了解是否有、以及哪个按钮被按下了。每一根行线和每一根列线都是一个组合，唯一代表着某个按键。一旦键盘微处理器发现有某个按键被按下，就向主机发送代表那个按键的二进制数据，也就是按键的代码，这一切都是在程序指令的控制下进行的。

需要说明的是，按键的二进制代码是以串行的方式送到主机的，也就是说，它把代表每个按键的二进制代码拆开，一个比特一个比特送到主板的键盘I/O接口，在那里，这些分散的比特将重新进行组装

对于不了解计算机内部的人来说，很容易想到键盘是直接连到中央处理器上的。理由很简单：可以通过键盘输入数字，这些数字可以直接被中央处理器接受，然后进行运算，甚至键盘上的0和1来直接编写二进制程序，计算机的运算过程就是这样实现的

这种认识是错误的，实际上，与所有I/O设备一样，键盘和主板上的I/O接口电路（键盘控制器）相连，在那里，键盘控制器与键盘互通信，接收键盘上按键的二进制代码保存在端口寄存器里，等等中央处理器取走。当然，它也会事先拍一下中央处理里的肩膀。

中央处理器并不认识键盘，实际上中央处理器谁也不认识，真正需

要键盘的是正在运行的程序指令。对于我们用户来说，恰恰是因为屏幕上显示一个窗口（可能是一个文字处理软件），需要输入什么东西，这时我们才会按一下按键，而与此同时，显示这个窗口的程序也正在等着你的按键，一旦它发现你按下了一个键，就会将其取走，如果你按下键盘的时候没有任何程序将它取走，键盘接口电路会丢掉按键代码，并发出警报。这就是为什么有时在按下键时会听到“滴滴”声

尽管所有的键盘都有0-9这十个按键，但并不意味着键盘或整个计算机机会把它们看成数字，换句话说，当按下2时候，键盘并不会发送一个二进制数字00000010，事实上，键盘上的所有按键都被当成字符看待，而不管它到底是0、9、A、Z或那些@！等符号。

20世纪50年代，人们用类似于现在传真机的终端连接到大型计算机，来分享它的计算能力，必要时还得借助于公共电话线路来传送数据。为此，1958年还发明了世界上第一个调制解调器，用于在电话线路（它原本只能传送像语音这样的模拟信号）上传送二进制数据

凡是需要多方参与、分式协作的事情，就必须有一个大家都能理解并共同遵守的标准。为了用同一种编码方法在各种不同的计算机设备之间传递数据，1967年出台了美国信息交换标准代码（American Standard Code for Information Interchange, ASCII）。这个编码方案是单字节的，最多只能表示256个字符。比如，00000111表示响铃，接到这个代码的设备应当发出一个脉冲使喇叭或蜂鸣器响一下，使人们能够意识到设备当前正处于什么状态；00000100用于通知接收设备，传输已经结束；00001010命令终端（电传打字机）另起一行，也就是换行。很明显，这些都用于控制数据传送过程，称为传输控制代码。

除了传输控制代码，这个标准里还定义了常用的字符，比如，00110000，也就是十进制48，代表0，依次类推，00110001（十进制49），代表1，00110010则表示2，字母A则是01000001（十进制65）

我们现今使用的键盘遵循这个标准，当然如何解释和处理键盘发送来的代码必须依靠正在中央处理器上执行的软件程序。比如，当我们熟悉的计算机器小程序运行时，它将接收我们的按键信息，并进行数学计算

假如要计算 $125+66$ ，这时，要先按1，2，5这三个按键，与此同时，因为计算器程序正在运行，它会从键盘接口分别取得三个代码00110001，00110010和00110101

但十进制数125对应的是二进制数01111101，而不是上面那一长串，要想利用中央处理器来做数学题，必须将那一长串代码转换成01111101，怎么转换呢？

键盘代码的安排很有规律，按键0的代码，其十进制是48；1的代码

是十进制的49.....这样“计算器”程序可以将从键盘接口那里得到的按键代码先减去48，得到一串二进制数，这样前面那三个代码先分别减去48，就得到了00000001，00000010，00000101，也就是十进制数1，2，5

但这依然是分散的3个数字，而不是我们需要的01111101（125），所以“计算器”程序将对这3个数字进行组装，方法是将它们依次乘以100，10和1，然后相加

$$1*100+2*10+5*1$$

通过这种方法，“计算器”程序就可以将3个按键代码转换成一个独立的二进制数01111101（125）

接下来，如果“计算器”程序发现你按了+键（代码00101011）就知道你要做加法，用同样的方法，它从你那里得到另一个数，然后做加法，整个过程就是这样，通过这个例子，看到键盘在计算机工作过程中的作用，并理解用键盘来做数学题并非我们想象中的那样简单直接

这意味着，键盘是为正在运行的软件服务的，而不是让你直接用按动0、1的方式来为计算机编写软件指令，尽管所有键盘上都有这两个键，当然可以通过键盘来编写程序，但需要另一个软件的帮助。据此我们可以判定，这样的软件在键盘发明之前就已经有了

## 14.5 显示器

键盘很重要，但如果需要一个地方来看看计算机的工作结果，这就需要一台显示器

显示器的历史可追溯到20世纪40年代之前。那时，为了打发业余时间，已经发明了电视机，但电子计算机刚刚起步，和现在的情况不同，当时，人们关心的是如何提高计算机的运算能力，至于计算结果以什么样的形式呈现出来相比不是那么重要。

第一次为计算机配上显示器的是在电视机发明几十年后，文字和图像更直观，没有理由让这么好的发明只是用来看电视

电视机的原理很简单，在讲述电子管的时候我们知道，真空状态下，灼热的阴极可以发射电子，电视机正是利用了这种原理，它的主体是一个喇叭形的玻璃管，称为显像管（抽成真空），如图14.9所示

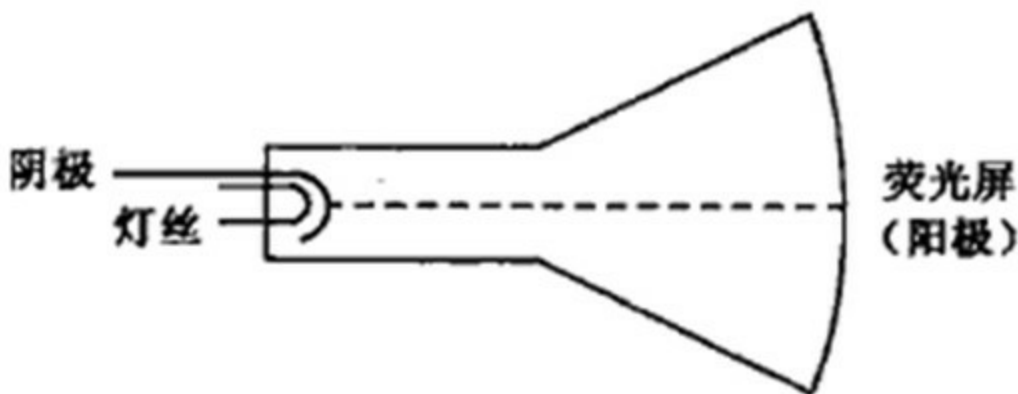


图14.9 显像管的大致构造

显像管的实际构造比这个复杂，首先需要额外的装置将电子聚集，变成细细的束；然后，显像管的前内壁上，也就是图中扇形的部分，涂有一层荧光膜，在电子束的轰击下可以发出光亮，发光的颜色与荧光粉的颜色有关，这是一个动能向光能和热能转化的过程，轰击的电子越多，速度越快，屏幕就越亮。所以，显像管还有一些必要的构造，用于加速电子的运动。

单纯是这样的构造只能在屏幕上显示出一个亮点。为了得到图像，我们需要在显像管外面套上两个线圈，分别叫行偏转线圈和场偏转线圈。这样，在两个线圈的控制下，电子束将一行一行从左向右、从上到下“扫描”屏幕，当扫描完一屏后，重新回到左上角，接着扫描下一屏

如果想在屏幕上显示一幅图像，很简单，只需要在扫描的过程中精确地控制电子束的有无就可以办到。被轰击的地方是一个亮点，没有被轰击的地方是一个黑点，于是整个屏幕就会显示出一幅黑白的画面

通常，我们看到的屏幕上会有好几百行扫描线，而一千行以上也不奇怪。不管有多少行，电子束都能又快又轻松地在一瞬间扫描完一屏——事实上，比我们印象中的“一瞬间”还要快不知道多少倍。就像灯泡断电后会熄灭一样，荧光粉只有在电子束的持续轰击下才会产生辉光，否则很快就会很快消失。所以要想让屏幕上的图像保持稳定，就必须以固定的频率不停地重复扫描它，这称为刷新，每秒的刷新次数称为刷新频率

上面讲的是电视机的成像原理，计算机的显示器与之相比并没有什么不同，但，和普通的电视机不同，计算机的显示器不需要声音，而图像也需要从空中或闭嘴电视信号线中接收。如图14.10所示，和电视机一样，当要显示一个字母符号时，所要做的仅仅是在电子束扫描到某个位置时，控制电子束的有无

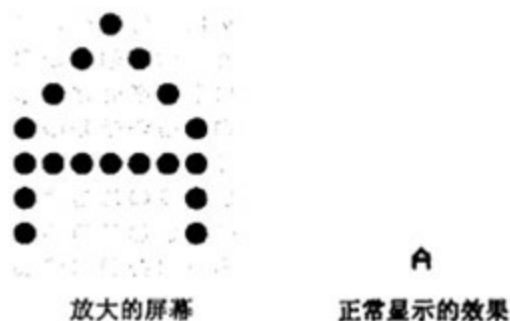


图14.10 显示器通过控制电子束的有无来呈现字符

电视机的声音和图像是实时的，与电视台同步，所以也就不需要任何存储设备。和电视机不同，为了在计算机显示器上产生稳定的图像，需要一块存储器储存所要显示的内容，这块存储器称为显示存储器，简称显存

显存通常位于负责显示图像的I/O接口中，一般来说是一块独立的接口卡（显卡），显存中的存储单元和屏幕上的每个像素一一对应。所以，如果显示器分辨率很高的话，可能需要一个容量很大的显存。对于显示器来说，最原始、最朴素的显示模式是黑白两色，那么显存中的每一位都对对应着显示器上的每一个像素，如图14.11所示

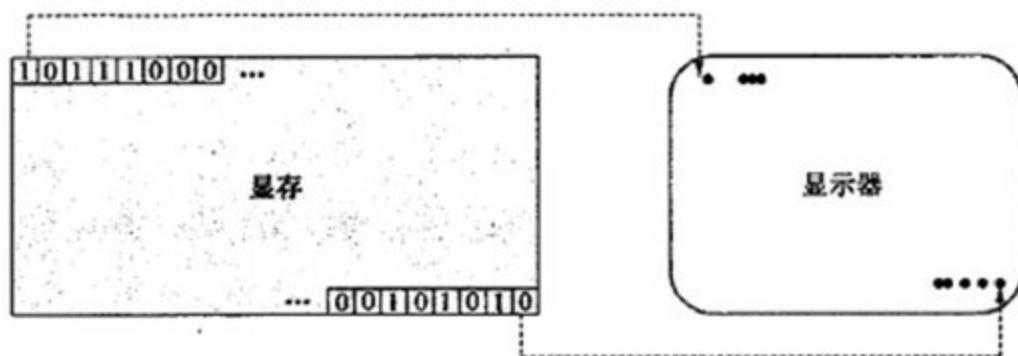


图14.11 单色显示原理：每个比特控制一个像素

要显示的内容可能来自任何地方，但毫无疑问地必须先由中央处理器通过执行指令来将它们搬运到显存里，比如，要显示U盘里的一幅图片，必须通过一个图片浏览程序将它从U盘读到内存中，然后，再以DMA的方式快速传送到显存。在这以后，中央处理器将不再过问这些数据，由I/O接口将这些像素数据通过信号线送到显示器。在那里，二进制像素数据被转换成模拟信号以及控制阴极的热电子发射，从而形成图像。

黑白两色用来显示文字还是不错的，但若是用来显示图片之类的东西，就有此恐怖，根本无法真实再现任何影像，除非像以前的黑白电视机一样，除了黑白之外，还能够显示一些灰色调。

想象一下，拿一瓶纯黑色的墨水，每当你向里面滴一滴牛奶后，它的颜色没有那么黑了。换句话说，每滴一滴牛奶，就会产生一种新的颜色，直到这瓶墨水变成纯白色（实际上不可能，因为无论怎样这里面还有墨水，只是人类的视力有限，所以这不是一个很好的例子）

从纯黑色到纯白色，这一系列的颜色称为灰度。取决于每一滴牛奶的量，如果恰好能用255滴牛奶使得墨汁从纯黑变成纯白，那么就能产生256种灰度，每一种颜色称为一个灰阶。当然，如果用了65535滴牛奶，那么将会产生65536种灰度，不过眼睛已无法分辨这么多种颜色

显示256种灰度，这对于显示器来说还真算不了什么，因为电子束的强弱可以非常直接地影响荧光粉的亮度和色彩。但，不能再像往常那样用1个比特来对应屏幕上的1个像素，相反，需要8个比特即1字节，因为1个字节可以表示十进制的0-255，当显示控制电路取得一个字节的灰度数据后，它会将其变成适当的电压提供给阴极，以控制它的强弱，从而使得屏幕上对应的像素呈现出相应的灰度，如图14.12所示

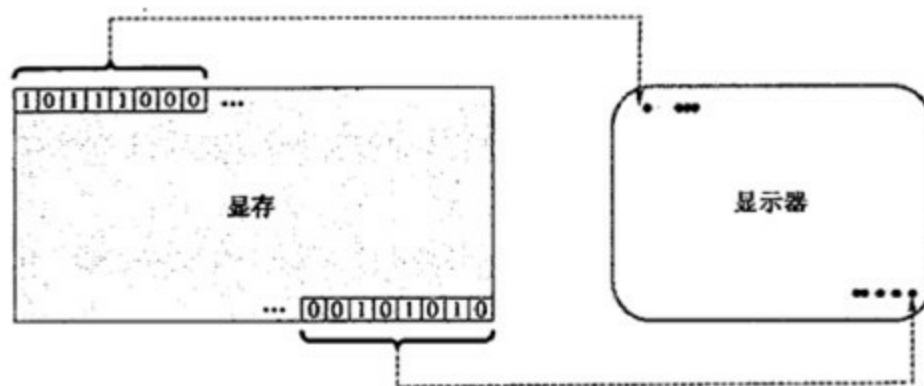


图14.12 灰度显示中每个字节控制一个像素

这意味着，在分辨率相同的前提下，256级灰度图像所需要的存储容量要比单色显示大8倍

在彩色电视机和彩色显示器出现之前，能够获得灰度图像也是令人

兴奋的，尽管不是五颜六色，但至少你能看清人物和风景的细节。时间在推移，什么也阻挡不了人们还原事物本色的冲动，就这样，彩色电视机和彩色显示器终于出现了

光是一种奇妙的东西，不可能不引起人类的注意，而且对它的研究也持续了好几千年。人们发现，尽管生活中有各种各样的颜色，但它们大都可用最基本的三种颜色调配而成，这三种颜色就是红、绿、蓝，称为三原色。取决于调配时这三种基色所占的比例，几乎可以得到任何一种我们想要的颜色。

掌握了不同颜色的调配方法之后，电子工程师们要在显示屏幕上制造调色板，用这种方法来使显示器产生丰富多彩的颜色。和单色显示器不同，彩色显示器的荧光屏用3个荧光点来共同组成一个像素，而不是从前的1个，如图14.13所示

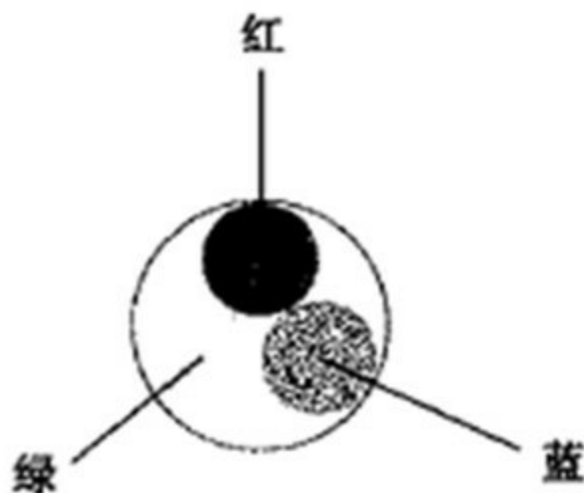


图14.13 彩色的显示原理

数量上的变化仅仅是一个方面，更关键的地方在于，这3个荧光点使用了不同的荧光物质，在电子束的轰击下会分别产生红、绿、蓝这3种不同的荧光。当然，电子束的强弱将直接影响到三原色的浓度。

彩色显示器必须使用三束电子流，以达到独立改变三原色调配比例的目的。可以从同一个电子枪分出三束电子流，并各自控制其强弱，也可以使用三个电子枪独立进行控制，事实上后一种技术用得更多，称为三枪三束显示器。

每个像素都包含红、绿、蓝3个荧光点，如果只有红色的荧光点被轰击而其余两个都没有，那么将显示一个红色的像素；如果要得到绿色或蓝色，可以照此办理。除此之外，要想得到其他各种各样的颜色，就必须恰当地控制这三束电子流。荧光点很小，产生的光线是散射的，而且它们又离的那么近，人眼得到的将是经过混合的光点，这就是它们调配出来的颜色



在单枪单束的时代，一切都很简单，可以用1个或几个比特，甚至1个字节来产生灰度图像，到了彩色时代，情况开始变得复杂，为了混合颜色，需要从显存里取得颜色数据，然后将它们转换成3路电流输出，分别控制3个电子枪

为了尽可能得到丰富的色彩，最好是红、绿、蓝三原色都能有256级渐变，就像从黑到白的256级灰度一样，这样，3种基本色都有256级色阶，那么它们就有 $256 \times 256 \times 256 = 16777216$ 种搭配，也就是16777216种颜色

相应的，为了能够在屏幕上显示16777216种颜色中的任何一种，屏幕上的每个色素要对应于显存中的3个字节，分别提供红色（R），绿色（G），蓝色（B）这3种色彩数据，当显示每个像素的时候，依次取出这3个字节，并将它们转换成适当的电流，分别控制3个电子板

显存中的颜色数据不是随意给出的，从这个意义上说，给出了确切的三原色数据，而显示器直接将其呈现在屏幕上，显示的是你真正想要的色彩，这称为真彩色。由于每个像素对应3个字节，共24位，所以也叫24位真彩色

现在是32位的时代，这是主流，而64位的计算机正在慢慢普及，你可能会觉得奇怪，32位或64位这和色彩有什么关系？

有关系，32位意味着什么？意味着中央处理器每次处理4个字节的数据，而且对存储器的读/写都是按每次4个字节进行的。当然，32位的计算机依然可以按每次一个字节，或每次两个字节访问存储器（毕竟是从8位、16位的时代走过来的，要向后兼容），但是要多费周折，严重影响计算机的速度，所以为了充分发挥32位计算机的性能，最好是把每次存/取的数据凑成4个字节，或4字节的倍数（好在存储器的价格便宜，要是从前也这么浪费，简直是罪过）。正是这个原因，现在流行的做法是采用4个字节来保存一个像素的数据，而不是理论上的3个字节，这也说明了我们的计算机可以设置成32位的原因，在这4个字节中，红、绿、蓝三原色数据各占一个字节，最后一个字节通常情况下不使用，把它置为0

65536种颜色是用两个字节来表示一个像素，称为16位色，因为每个字母最大可表示65536

对于32位色来说，每种原色有256级过滤是非常自然的。但16位色只用两个字节来容纳三原色数据，这样平均下来每种原色只能有32级过渡，使得这种渐变对肉眼来说具有跳跃感，图像质量的显示效果大打折扣

传统上采用电子束成像的显示器称为阴极射线管显示器，即CRT（Cathode Ray Tube）显示器，第一次在电子计算机上使用CRT显

显示器是在1949年。CRT成像技术有很多优点，比如色彩艳丽、图像清晰、反应灵敏，但缺点是体积比较大，笨重、耗电量高，所以对一般的应用而言，这几年液晶显示器开始流行起来

液晶显示技术在光学上的CRT是一样的，都是利用三原色混合原理，但，为了混合色彩，液晶显示技术使用了不同的材料

1888年，奥地利植物学家莱尼茨在工作中发现，某些有机物，如胆甾醇的苯甲酸脂和醋酸脂（）熔化后，会经历一个不透明的白色浑浊液体状态，并发出多彩而美丽的珍珠光泽，在继续加热到某个温度后，又会变成透明清亮的液体

这当然是很奇特的，第二年，一位名叫莱曼的德国物理学家开始研究这种现象，想从微观上看看这里面都有什么奥妙。他还设计了一款最新的、带有加热装置的偏光显微镜。他发现了一种具有晶体性质的特殊液体，并将其命名为液态晶体（Liquid Crystal），简称液晶

物质有三态：固态、液态、气态。之所以会有这三种状态，很大一部分原因在于组成它们的原子和分子间距不同。在前面讲半导体时说起过晶体，当然我们指的是固态晶体，这些东西都具有某些共同的特点，比如原子或分子排列很规整，具有固定的熔点，而且在光线的照射下会呈现晶莹的光

不单是固态的晶体，液态的东西里也有晶体，换句话说，有些液体，组成它的分子具有按一个方向整齐排列的特点，以至于它具有晶体的性质，这就是液晶。如图14.15所示，液晶分子大体上都呈细长棒状或扁平片状，长约10纳米，宽约1纳米

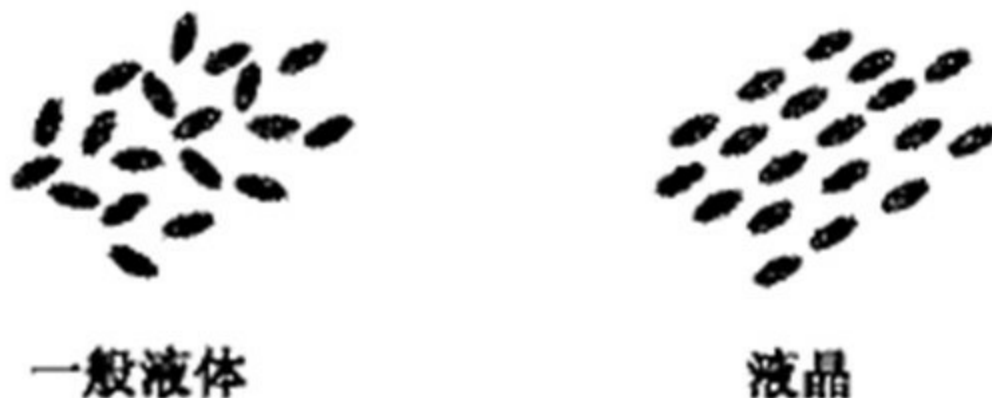


图14.15 液晶的形态

这或许算不上什么，但神奇的是，将液晶放在两个极板之间，在两端加上电压后，这些分子马上会整齐地改变排列方向，如图14.16所示

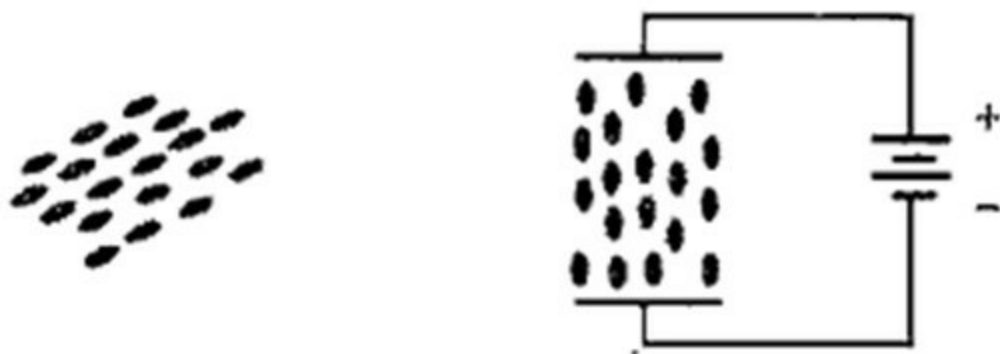


图14.16 外加电压可使液晶的排列发生改变

实际上，早在发现液晶之前，也就是19世纪初，科学家们就开始研究晶体的性质。那些常见的性质，比如固定的熔点、美丽的光泽、规则的外观就不用说了，神奇的是，他们还发现很多晶体物质可以使光线发生旋转，也就是旋光性。什么是旋光性呢？

研究液晶的旋光性需要一个偏光片，如图14.17所示，注意偏光片上有一条透光的沟槽



图14.17 偏光片

很早以前麦克斯韦已经证明了光也是电磁波——沿各个方向振动的波，通常，自然光来自各个方向，有的直接来自我们注视的物体，这是需要的；而另一些则来自其他方向的反射和散射光，这会干扰我们的视线，大晴天的时候，前方明晃晃地看不清东西，就是这个道理，而偏光片的作用是可以滤掉反射和散射光，留下那些振动方向和沟槽一致的光线

现在将灯泡放在偏光片的背面，通电后将灼热发光，这将在偏光片的另一面产生一条扁平的光带——就像白天黑暗小屋中窗帘缝隙透过的阳光一样，这对于任何一个人来说都是非常自然和容易理解的。要在你在这束光前进的途中放置一个透明的东西，比如一块玻璃，光线照样能穿

透它，不受任何阻碍地继续前进，就好像这些障碍物根本不存在，如图14.18所示

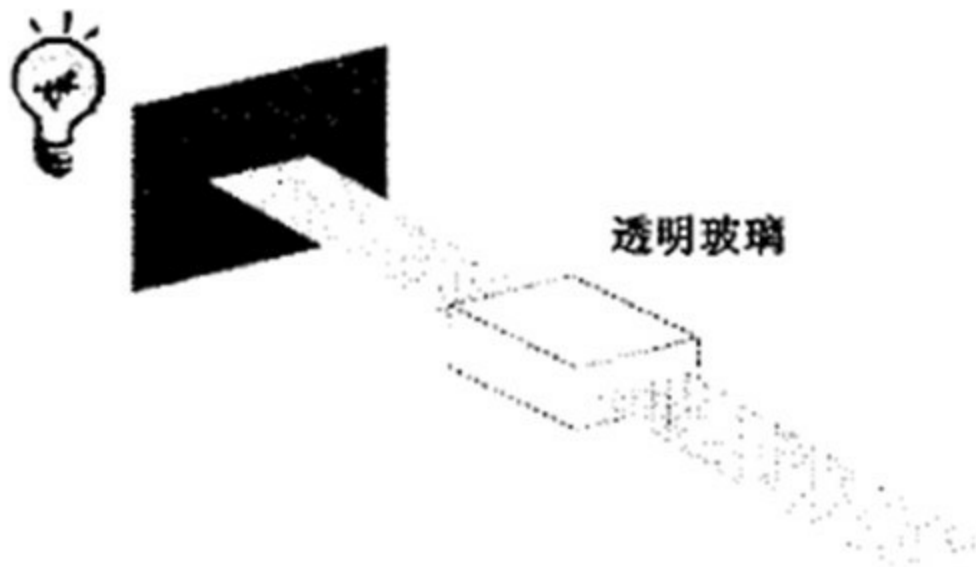


图14.18 光线穿过玻璃时并不会有任何变化

玻璃不是晶体，它没有固定的熔点，要是把玻璃拿开，换上某种晶体，奇怪的事情发生了，光线居然在通过晶体时放置了一个角度，如图14.19所示

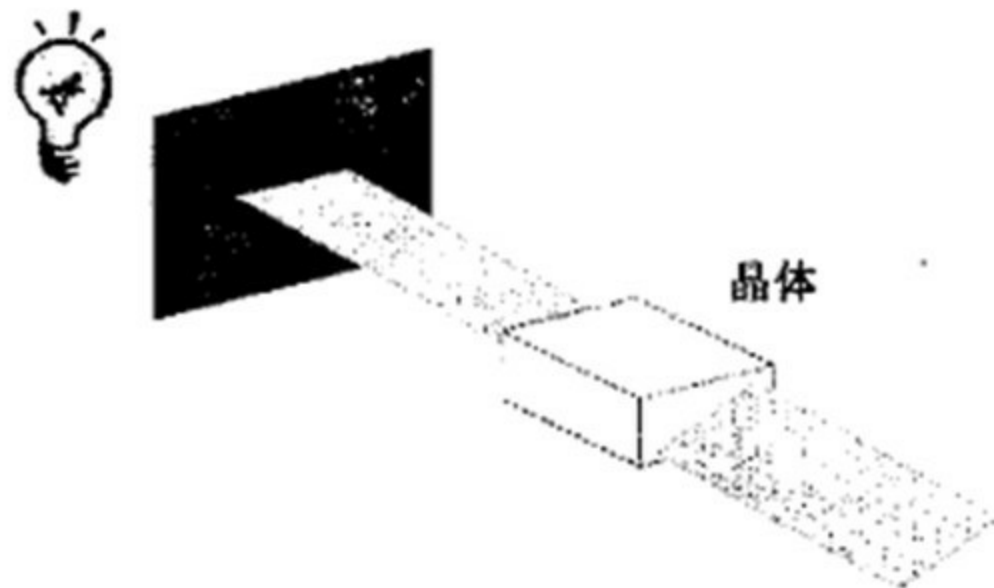


图14.19 光线穿过晶体会发生旋光现象

这就是所谓的旋光性。有很多晶体具有使光线放置的特性，这里面也包括液晶。为了利用液晶成像，需要准备两个偏光片，但它们的沟槽呈不再的角度，如图14.20所示



图14.20 用液晶显示一个像素需要两个偏光片

在这两个偏光片之间放置一个液晶分子，这样当光线从一个偏光片的沟槽照射进来后，经液晶分子旋转，正好能从另一个偏光片的沟槽中穿出，这时，我们就能看到亮光，如图14.21所示

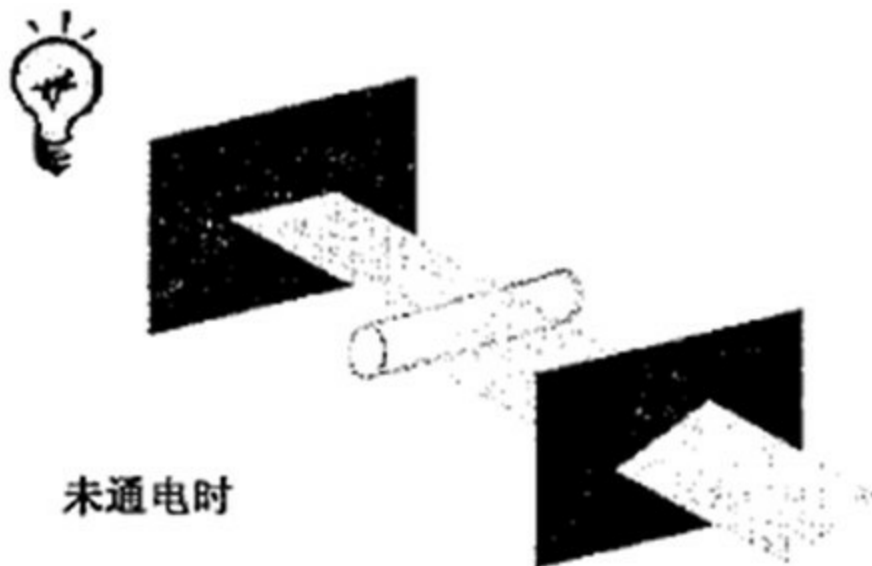


图14.21 当没有外加电压时，光线可穿过两个偏光片

如果仅仅是想让光从这两片偏光片的沟槽里穿过，大可不必这么麻烦，我们需要一点点改变，期望能因此而带来令人惊讶的效果。现在，我们将在两边加一个适当的电压（通常的做法是在两个偏光片上各加一层导电玻璃，并将它们分别接到电源的正、负极上），由于液晶分子立

即改变排列方式，光线将会不经旋转地直接照射在另一个偏光片上，又因为两个偏光片的沟槽并非平行，而是呈一个角度，所以光线无法穿过第二个偏光片的沟槽，于是看不到亮光，而是黑色，如图14.22所示

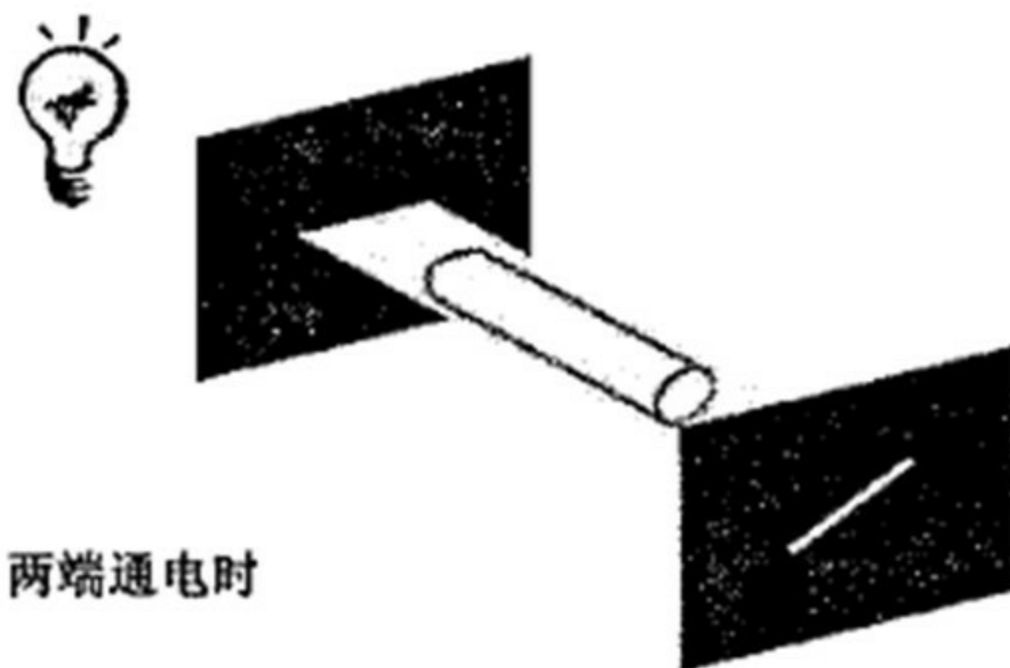


图14.22 当有外加电压时，光线消失

很巧妙吧！这就是液晶显示器的显示原理。说明了它是如何显示单个像素的，当然，在这个例子中还只能显示黑白两种颜色，要想显示彩色，则需要三个这样的液晶构造，通过加上一层滤色膜，就可以让它们分别显示红、绿、蓝三种颜色，并混合在一起呈现某种彩色，在这方面，它和传统的CRT显示器使用了相同的原理

这只是一个简单的示意图，液晶显示器的实际构造远比这个复杂。一台液晶显示器里密密麻麻分布着几百万个这样的像素构造体，光是如何巧妙地安装和连线就让人惊叹了

前面我们重点说了显示器的成像原理。在物质准备好了之后，剩下的问题是如何在显存中布置我们所要显示的图像呢？

这里有几种可能，第一种前面已经说过了，那就是简单地把要显示的内容快速地写进显存，显卡总是在不停地读显存并加以显示，显存里的内容更新的有多快，屏幕上的图像就变换得有多快

第二种是动态生成的，比如要在屏幕上画一条直线或一个圆，按常规方法，需要事先将组成这些图形的每一个点保存起来，下次显示的时候再原样写入显存的相关单元，就像保存和显示一幅照片那样。不过，利用几何知识，只需要给出一条直线首尾两个端点的坐标，或者圆心位置和半径，直线或圆上的其他点都可以通过数学公式计算出来

所以，这样就大大减轻了编写程序的负担，每次只需要指定一些参

数，计算机就能自动得到其他点在屏幕上的位置，并自动填充显存。非但如此，其他一些显示效果，比如半透明、渐变、暗化、亮化、图像填充、动画过程等等都可以自动完成。

不管采用哪种方式在显存中布置图像，遗憾的是存储器一直不够快，所以要想快速显示动画，传统的方法是将显存分成几个部分，称为位平面，每次只把一个位平面上的内容呈现在显示器上。在一个位平面正在被显示的同时，另一个位平面也在快速地填充，然后瞬间切换，于是就看到了无比流畅的运动图像。

以前上面所说的这些计算任务都是由中央处理器负担的，中央处理器很忙，干这些事情会很吃力，所以现在都交给了显卡上的微处理器，称为图像处理器（Graphic Processing Unit, GPU），实际上也就是数字信号处理器DSP，在我们现在的个人计算机上，GPU的复杂程度不亚于中央处理器

现在图形图像已经成了一个热闹领域，动漫制作、平面设计也已经成为一种行业，并且正在吸引着无数的人从事这方面的工作

# 第15章 计算机的启动过程和操作系统

在电子计算机如此普及的时代，它已经和普通意义上的家用电器没有什么区别了。



## 15.1 打开电源并启动计算机

在计算机发明到现在的几十年间，所有和计算机相关的东西都变了，变得更复杂、更强大，唯一没有变的就是当计算机开始工作时，内存里必须已经放好了要执行的程序指令。而为了做这项准备工作，那个时候人们曾经用过机械开关，依靠这种笨拙的手段来一个地址一个地址地填充内存里的空间。

但人们很快意识到这并不是一个好办法。在编写了大量、用于做各种不同工作的程序指令后，要想重复使用它们，最好是永久地保存起来，下次要用的时候再快速地装载到存储器里。这样，在电子管和晶体管时代，纸带和穿孔卡片成了不二之选

所谓纸带，就是一卷长长的纸条，用手工或机器把二进制数据刻在上面，有孔代表1，没有孔代表0，如图15.1所示

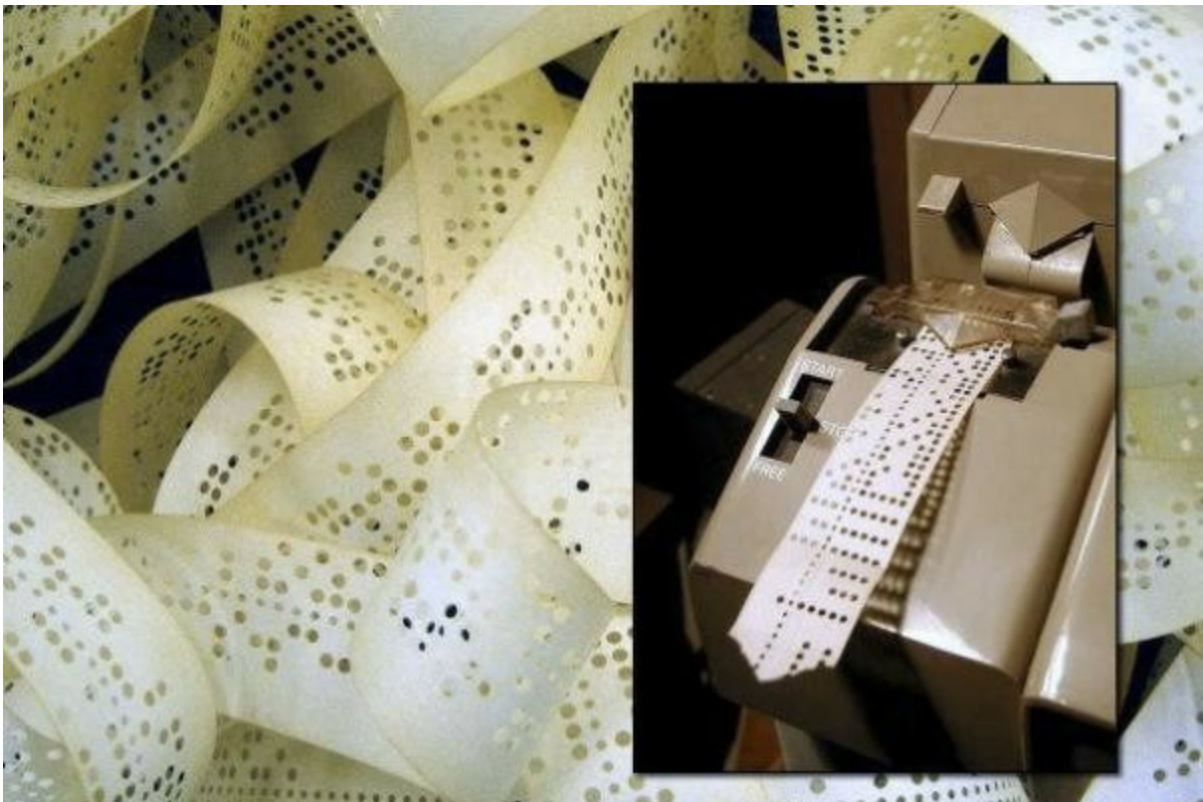


图15.1 纸带

除了纸带，还有穿孔卡片，不同之处在于后者是一张一张的，在纸上打孔看起来很简单，但每一排孔代表的是一串二进制比特，而每一串二进制比特又是一条特定的计算机指令，指令是不能出错的，所以在纸上打孔要细心。在这个过程中难免会出错，不该打眼的地方打了眼，就得拿胶水、剪刀、纸带屑、补孔板修补

一般来说，每一卷纸都用于完成某个特定的计算任务，平时放在仓库里，下次要执行相同的任务就方便了。由于程序指令必须安排到存储器里才能执行，所以还需要使用纸带阅读机来识别这些孔，当纸带前进的时候，纸带阅读机就一行一行地把它转换成二进制数，并写入存储器

上面所说的是输入，对于输出，也采用相同的方法，由打孔机或穿孔机把计算结果打成一排一排的孔，在大型计算机时代，政府和企业只能排除使用这些机器，把纸带送进去，然后等候，就像等待神谕一般，当机器计算出了结果，也把它以穿孔纸带的形式呈现。

很明显，如今的情况完全不同了。问题是我们现在的计算机体积上小了很多，既没有纸带也没有见谁在纸带上穿孔，打开计算机电源，我们什么也没做，内存中空空如也，它怎么居然就能自己开始运转了呢？

不要被表面现象所蒙蔽，除非你压根就没往这方面想过。这里有两条好消息：第一条是我们以前学过的知识没过时，中央处理器仍然和内存直接打交道，而内存也还是那么健忘；第二条是，就我们目前的计算机体系结构而言，中央处理器不但和内存相连，还和一个只读存储器ROM相连，这实际上是把中央处理器可以访问的地址空间分成了两部分，比如，如果访问00000000-01111111之间的任何一个地址，那么访问的内容就落在内存之中；但如果访问的地址落在10000000-11111111之间，那么实际上访问的就是ROM里的内容（图15.2）

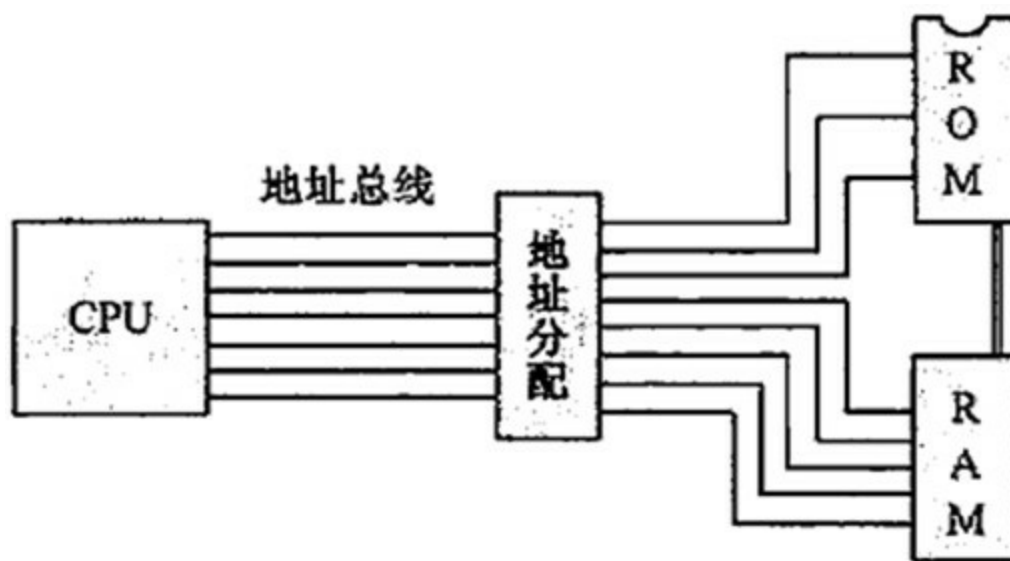


图15.2 中央处理器的地址空间分配

注意，内存和只读存储器是两样东西，ROM位于主板上，是主板在生产时就有的一部分

每次打开电源后，如果中央处理器是好的，它就开始工作，从存储器里取指令，然后执行。工程师在设计中央处理器时动了一番脑筋，让

它第一次开始取指令执行的地址恰好位于ROM之内，而ROM在制造时已经固化了一些指令，这使得中央处理器不会因为存储器还没有准备好指令而不知道做什么

ROM中固化的内容挺多，但有相当一部分是用来对计算机的各个部件进行检测的指令（包括中央处理器自己的一些内部零件），看看它们是否完好无损；另外，还要让它们调整一下工作状态，这叫做初始化，这一切都是很有必要的，要是在正式开始干活的时候出故障就不好了

除了加电自检指令，ROM中还包含一些访问外部设备的指令，我们可以用这些指令来访问设备而不必亲自编写，因为我们可能不熟悉如何与这些设备打交道。传统上，这就是设备驱动程序。不过，它包含的只是少量常用的外部设备，比如键盘、打印机、显示器和后面要提到的硬盘，对于一台计算机来说，要想正常工作，它们是最基本、最低限度的配置。同时，这些设备驱动程序仅提供最基本的功能，以保守的方式使用对应的外部设备。正因为如此，该ROM更多地被称为基本输入/输出系统(Basic Input/Output System, BIOS)

整个检测和初始化过程只有短短的几秒钟，但过程很复杂。传统上，这个过程叫做POST（Power ON Self Test,加电自检），从CPU开始，首先检测主板上各个电路模块，如果发现问题，就通过喇叭发出不同的单调表示，如果机器连喇叭都坏掉了，就只能通过仪器进行测试。这是一个约定，中央处理器向某个约定的端口发送表示出错的二进制数字代码，维修人员可以在那里通过仪器获得二进制数字代码。

尽管看起来通过屏幕显示出了什么问题可能更直观，但此时显示器还没有初始化，初始化的工作就是检测显示器I/O接口是否存在（更通俗点的说法就是显卡是不理已经插在主板的扩展槽上了），要是存在，就让显卡进入正常的显示状态，特别是，显卡还要准备一个原始的驱动程序以供使用。显示部分的检查工作不是太靠前的，不过很快，显示器开始工作了

当显示器开始显示内容时，就会显示检测和初始化过程，但不是所有的检测工作都在屏幕上执行，除非检测到了问题。注意在这个时候，会看到中央处理器的类型和内存大小，当检测内存时，用的是一种笨方法——向每个存储单元写入一个数，再读出来，看是不是一样（这是个笨方法，但很有效）

加电自检像一场盛大的检阅活动，几乎所有的部件都参与其中。下次当你看到键盘上的指示灯闪烁几下，或听到打印机怪叫，表示这些设备正在接受中央处理器的检阅

加电自检或叫上电自检，无疑是非常非常重要的，但是执行的这些

程序指令仅仅是为了检验计算机硬件是否健康，跟我们真正想要做的事情没有一点关系。我们使用计算机的目的是希望它能运行我们自己的程序指令，帮助我们解决各种实际问题，那么，我们的程序指令在哪儿呢？计算机怎样才开始执行它们呢？

答案不在ROM里，因为它的容量通常很小，即使ROM容量大，也不可能制造的时候就能预知你要用哪些软件，同理，答案也不在内存里，更何况内存无法在停电的时候持久保存内容。软件程序的更新换代很快，所以答案是：那些可以随时更新内容的大容量存储设备，或者叫辅助存储设备，如硬盘、光盘等

## 15.2 各种各样的辅助存储设备

计算机工程师们用纸带来记录程序指令可能是有道理的，尽管在计算机内部，程序指令是一系列电信号，纸带是这些电信号的非常直观的写照，但很快纸带就被淘汰了，取而代之的是磁记录技术

磁玄妙不是为了电子计算机而发明的，相反它最初的目的是把声音保存下来而发明的。声音从本质上说是一种振动。1877年，考虑到这种振动可以连续地记录在一个圆盘或滚筒上，爱迪生发明了留声机。留声机可以用手摇驱动，也可以用电动机驱动，而圆盘表面是一层锡纸、蜡或虫胶。录音时，声波使一根针在圆盘表面留下深深浅浅的坑。同样是这根针，播放的时候深深浅浅的坑让它以不同的幅度推动一个纸片，纸片使空气振动，就听到保存的声音了。

美中不足的是，留声机刚发明出来的时候，声音太小，因为录音时，针在圆盘上留下的坑太浅，播放时这些坑引起的空气振动就不强。直到后来出现了电子管和晶体管，才改变了这种局面

之后人们又发明了钢丝录音机。因为当时已经有了电话，而且人们掌握了用电子管把信号放大的技术，人们把细钢丝绕起来，在电动机的牵引下运动。钢丝上有一个电磁铁，在这时被称为磁头，当通上音频电流时，会在钢丝上留下剩磁，随着音频电流强弱的变化，在钢丝上留下的微小磁场也会有强有弱，等于把声音信号记录下来了。播放时，钢丝上的剩磁会在磁头上感应出微弱的电流，经过电子管的放大，使扬声器振动，如图15.3所示



图15.3 传统的录、放音原理

钢丝录音机的音质并不好，而且在当时又十分昂贵，所以并没有像它的发明者所期望的那样流行和普及。相反，有一种新型的替代品——磁带走进了人们的生活

磁带录音机的原理和结构与钢丝录音机非常相似，只是用的是磁带——用塑料制成的带子，表面涂有一层磁性材料，如三氧化二铁或氧化铬，在电动机的牵引下经过磁头来记录或重放声音。

就在大批科学家忙着将声音和图像录制在钢丝、磁带上时，大

批计算机学者们正在给电子计算机纸带穿孔。他们发现了磁带正是他们所需要的，他们认为，用磁带来记录计算机的数据和指令，会给他们带来前所未有的方便。于是，磁带顺理成章地成了当时那些大型计算机的重要外部存储设备。不过使用磁带有一个不很方便的地方，那就是在磁带上寻找想要的信息很不直接，因为磁带是绕成卷的，如果想要的数据恰好在磁带的末端，还得快进一会儿才能找到它。换句话说，磁带是顺序检索的

最开始这能将就，但随着计算机在各个行业的应用越来越广泛，自然对访问速度提出了更高的要求，于是1956年9月13日，雷诺·约翰逊和他的同事们首次将一个叫硬盘的东西安装到了计算机中，这宝贝足有两个冰箱那么大，重达1吨，从侧面看上去像一把巨大的梳子，容量只有4.4MB

在当年约翰逊带队的硬盘小组中，艾伦·舒加特后来发明了历史上第一块双面硬盘。1973年，艾伦·舒加特离开原来的公司和团队，成立了自己的公司，专门从事软盘驱动器、硬盘驱动器的研发和制造。他的公司曾经主导了整个硬盘产业的快速发展，在短短的几十年里使硬盘无论从体积还是容量乃至速度都发生了翻天覆地的变化，成为全球增长最为迅猛的计算机产品。

硬盘的构造并不复杂，但毫无疑问非常精密，它由一个以上的盘片组成，构成盘片的基本材料一般是铝，上面镀一层磁性材料，这也是它之所以被称为硬盘的原因。这几个盘片由一根轴，在电动机的驱动下高速旋转

为了在硬盘上读/写数据，每个盘片都需要两个磁头 – 上、下各一个，像普通录音机一样，二进制的0和1被转化成两种强度不同的磁场，通过磁头记录在盘片上

磁带是顺序存取设备，要在它上面读程序和数据，通常只能从头到尾慢慢地找。为了快速访问程序和数据，硬盘提供了更好的解决方案（不一定是完美的解决方案 – 历史证明了这一点）

首先，硬盘盘片是圆的，磁头位于它的表面，可以将转动着的盘片表面磁化，通过这样的方式来记录数据。这意味着，在盘片上写数据的时候，会在转动着的盘片表面形成一个圆形的磁化区域，这称为磁道

其次磁头也是可以移动的，从盘片的中心到边缘，或从边缘到中心，这样磁盘表面就不止一条磁道，尽管看不见磁道，但可以想象它们将在盘片上形成密密麻麻的同心圆。为了读/写某条磁道上的数据，磁头可以快速移动到那里，仅凭这一点就和磁带有了本质上的区别

现在的硬盘都不大，如图15.4所示



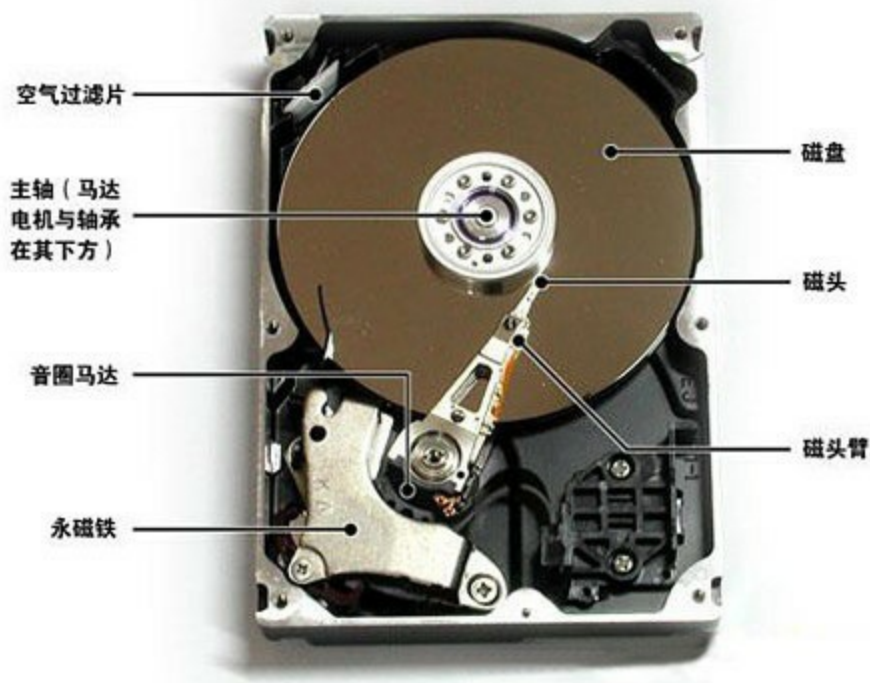


图15.4 硬盘的内部构造

这是一个拆开的硬盘

硬盘的盘片从0开始顺序编号，第一个盘片的正面是0，反面是1，第二个盘片的正面是2，反面是3.....由于每个盘面都有一个磁头，所以通常不说“盘面号”，而称之为“磁头号”。在每一个盘面上，各个磁道也从0开始有一个顺序编号，也就是磁道号，现在想象一下，在所有盘面上，位置相同的磁道会共同形成一个圆柱，这就是柱面

“柱面”这个概念的形成是有原因的，在多数人的意识里，硬盘记录数据的方式是以盘面为单位的，一个盘满了，再使用另一个盘面，实际上，这并非是一种高效的工作方式，读/写硬盘时，移动磁头寻找磁道（寻道）是一个需要浪费大量时间的机械动作。为了加快硬盘的读/写速度，如果一个磁道容纳不下的话，最好是保持磁头不动，而把数据分散到各个盘面的同一磁道，换句话说，柱面是硬盘读/写的一个基本策略

磁道还不是硬盘读/写的最小单位，事实上，磁道进一步被划分成细小的片断，由于磁道是个圆环，所以这些片断也呈弧形，称为扇区，在扇区与扇区之间，有一个小小的间隔，用来减小扇区之间的干扰，可以想到，基于相同的原因，磁道之间也会有一个小小的间隔

每个扇区以一个扇区头开始，在这个区域里标记了该扇区所在的磁头号、磁道编号以及自己的编号。另外，在扇区头里还有一个二进制标记，表明该扇区是不是因为物理损坏而不能使用

扇区头还有其他内容，但与我们的话题无关了。接下来，对于每个扇区来说，真正用于存储用户数据的地方是在扇区头之后，一般有512字节

硬盘属于旋转设备，磁头只能在盘片的半径方向往复运动，对磁道的读/写必须依靠盘片的高速旋转。盘片的转速非常高，每分钟3600圈（圈/分钟表示为r/min，即rounds per minute，也称为转/分钟）现在被认为很低了，这个速度目前可达到7200圈到10000圈，通常认为较快的转速对于缩短硬盘的读/写时间是有利的

和所有连接在计算机上的东西一样，硬盘，包括后面要讲到的其他辅助存储设备都属于I/O设备，有自己的I/O接口电路，换句话说，它们都有自己的I/O端口，使用中断和中央处理器通信。同时，因为存储设备总是和大量的数据相关，所以毫无疑问地会采用DMA机制

存储设备的I/O接口不是一成不变的，因为我们总希望它们的数据传输速度能更快，要知道，尽管中央处理器一直对内存的速度感到不满，但硬盘的速度实际比内存的还要慢10000-100000倍。为了提高内存和辅助存储设备之间的数据吞吐量（每秒钟传输的比特数或字节数），人们发明了不同的I/O接口，比如先进技术外设（Advanced Technology Attachment, ATA，习惯上称为IDE-Integrated Drive Electronics，电子集成驱动器，本意是指控制器与盘体集成在一起的硬盘驱动器）、串行先进技术外设（Serial Advanced Technology Attachment, SATA）、小型计算机系统接口（Small Computer System Interface, SCSI）等，这还不是全部。各种不同的接口有自己独特的地方，特别是接口电路和硬盘实体之间的连线形式（决定连线形式的是它们内部迥然不同的工作机制）。比如ATA是并行传输，SATA是串行传输，但说来说去，人们最关心的其实还是数据传输速度

硬盘读/写的基本单位是扇区，要做到这一点，那些希望在硬盘上写数据或从磁盘上读取数据的程序必须通过中央处理器给硬盘I/O接口发出指令，把磁头号、柱面号、扇区号以及数据在内存中的地址等，告诉I/O接口的端口寄存器，这样，剩下的工作就由硬盘来自动完成。

世界上第一个光盘产品诞生于20世纪70年代，它的主要原理是用激光的两种不同反射状态来记录二进制数据。光盘记录数据的主要材料是能够在激光的照射下改变状态的化学材料，以及位于其后面的反射层，通过控制大功率激光束的有无，可达到使某些地方的化学材料透光性变差，而另一些没有变化的目的。以后要读取这张光盘时，根据反射光线的强弱有无来还原这些数据。

存储技术的竞争通常集中在两个方面：容量和成本。20世纪90年代，人们普遍认为硬盘的记录密度很难再进一步提高。就在人们期盼存



储领域里的新发明时，硬盘又控制了局面。当然这是个好消息，保护了投资。传统的硬盘和老式磁带录音机一样，用电磁感应的原理工作，但随着比特密度的提高，每个比特的磁场越来越弱，读取越来越困难，更不要说为了进一步提高密度。后来又发明了一种方法，将每个比特的磁场立起来，像钉子一样楔入磁层中，这就是垂直记录技术。

1988年，法国人阿尔贝·费尔和德国人彼得·格林贝尔分别发现，有些东西的电阻会受到磁场的影响，非常羡慕虚荣磁场变化就能显著改变它们的电阻（从而显著地改变电流），这一发现巨磁阻效应。1994年，第一块采用巨磁阻磁头的硬盘诞生，记录密度一下提高了十几倍，容量从4GB提高到了600GB甚至更高。采用巨磁阻技术的硬盘使用两种磁头，写入时还是采用电磁感应方式，而在读的时候采用巨磁阻磁头

2007年，阿尔贝·费尔和彼得·格林贝尔获得了诺贝尔物理学奖

光盘有自己的优势，比如便于随身携带，或保存暂时不同的数据，后一种情况使得它更适合用于备份，即把目前用不到的数据记录在光盘上，并存放到安全的地方以备不时之需，同时将其硬盘空间腾出来。总的看来，光盘与其说是取代硬盘，不如说它是硬盘的有益补充，至少目前是这样。

另一种看来有希望从传统硬盘那里夺走权杖的是固态磁盘，它出现于20世纪90年代，世纪末，半导体材料的研究和集成电路制造技术都获得了突破性进展，固态磁盘恰恰是这两者相结合的产物。我们平时所使用的U盘和存储卡以及我们现在说的固态磁盘，它们使用的材料相同，但属于不同的技术。如图15.5所示，固态磁盘可以做得很轻很薄，因为它内部只有集成电路芯片而没有机械旋转部件



图15.5 固态硬盘

固态硬盘独特地方是质量很小，不需要电动机和盘片，没有磁头，更不要说旋转。要是它掉在地上磕碰了一下，唯一可担心的就是有没有沾上灰尘，但传统的硬盘不同，当它受到磕碰时，磁头可能会划伤盘面，导致坏块或磁盘损坏

和固态硬盘不同，U盘不使用IDE或SATA之类的I/O接口，相反它

使用USB接口。所以它们工作方式是不同的。U盘的发明者是哈尔滨朗科科技有限公司，这是几十年来，中国在计算机存储领域唯一的原创发明专利成果。

但目前固态硬盘仍处于劣势，原因很简单，无非就是容量、性能和价格。硬盘可提供几兆、几十兆字节的容量，注意20世纪90年代初能够买到1G字节容量的硬盘会令人欣喜不已，但现在，硬盘容量可达到几TB。固态硬盘在容量上还无法与传统硬盘匹敌，而且制造成本也一直居高不下，短时间内要想撼动传统硬盘的统治地位还不太现实

传统硬盘是否已日渐式微，固态硬盘是否最终最终胜出，还很难说按道理，在计算机启动之前，用户自己的程序指令应该已经在硬盘上准备好了，当计算机完成自检后，它应该从硬盘中把用户的程序读入内存，然后用一个跳转指令转到用户程序的开头继续执行，以完成用户自己的工作

想法是不错，技术上也是可行的。问题是：硬盘那么大，可能存储了几个、几十个、几百个程序，有的用于办公，比如文字处理和电子表格，有的用于浏览见面，有的用于娱乐游戏.....这么多的程序，计算机怎么知道我要用的是哪一个呢？或者说，要在这个过程中体现用户自己的意志，如何插手呢？如果计算机是一家餐馆，那么这个过程可以理解为拿出菜单让顾客点菜

解决这个问题最好的方法是将它们都 显示出来，允许用户从中挑选一个，这意味着，有责任告诉使用计算机的人硬盘上有哪些程序，并允许他们选择。还有，要是硬盘上的程序和数据不中意不需要了，应该可以删除它们。如果硬盘有足够的地方，应当允许添加新的程序和数据

要达到这些目的，单纯指望硬件比如中央处理器是不现实的，因为中央处理器只知道执行有限的那些指令，所以要完成这样的任务就需要另外编制一种程序，这种程序本质上和其他用于文字处理、电子表格、上网、听音乐、玩游戏的程序没有什么两样，但专门用来管理和其他辅助存储设备上的东西，将它们显示在屏幕上，并负责按你的要求添加新的、删除不再需要的，甚至帮你将这些程序和数据交给中央处理器去执行

这种程序可以单独编制，独立存在，但事实上它经常是另一种程序的组成部分。“另一种程序”更大、更复杂，做的事情更多，这就是操作系统，操作系统的能耐我们慢慢了解，现在最关键的是，当你打开计算机电源并完成自检过程后，操作系统将是中央处理器第一个要召见的对象

## 15.3 启动操作系统

“操作系统”这个名称的由来和它在人与机器之间的地位有关。计算机只是一种工具，当然是非常高级、非常先进的工具，但人的需求更重要。没有人用计算机，就不会有市场需求，也不会有那么多公司投入那么多的金钱，用那么尖端的设备来进行研发。同样，为了使操作计算机变得更简单、更容易，也就有了这种特殊的软件，像社会上的各种服务代理机构一样，让你快速、便捷地办理各种事务，这就是为什么称它为操作系统的原因

很明显，操作系统也是人工编制的程序指令，它的历史几乎和电子计算机本身的历史一样长，当然都只有几十年而已。曾经出现过的操作系统有多少种，现在已难以统计，它们由于各种各样的目的用在各种各样的计算机上，这也是它不能固化在ROM中，而必须单独存在的原因之一。在这些种类繁多的操作系统中，有几种是大家用得，很熟悉的

（有的仅仅是名字让大家觉得熟悉），如UNIX、Mac OS、Windows等

一般地，允许在一台计算机上安装多个不同的操作系统，特别是考虑到每一种操作系统都有自己的特色和吸引人的地方，也许还有一些特殊的功能。即使只有一块硬盘，现有的技术条件也允许你实现这个愿望。

注意尽管我们下面重点以硬盘为基础来讨论问题，但同样也适用于最新的固态磁盘。为了保证技术的延续性，保护软件/硬件投资，固态磁盘在内部将自己模拟成可以按扇区来读/写；在外部，继续同传统的硬盘接口保持一致

如果确信要在一块硬盘上安装多个操作系统，那么首先要将硬盘分区

中央处理器在开机之后，首先从ROM中取得指令开始执行，借此完成硬件检测和系统初始的布置工作，接着，中央处理器把硬盘0面0磁道第一个扇区的内容读入内存，然后用一条跳转指令进入这里继续执行

硬盘的0面0道第一扇区，称为主引导扇区，这是一个约定。主引导扇区有512字节，包含了446字节的启动指令和数据，后面是64字节的分区表，最后2字节必须是01010101和10101010，表明主引导扇区是有效的，可用的。注意这2字节的特点，它们都是由0和1交替组成的，称为“花码”

分区表指明当前这块硬盘被分成了几个部分，总共允许有4个主分区，每个分区的资料16字节，指明该分区的起始位置和大小、类型以及是否为活动分区。原则上类型是一个二进制数，用以表明该分区是由哪

种操作系统负责。因为每次只能启动一个操作系统，所以活动分区的意思是“可启动的”或“应该被启动的”。4个主分区中，只允许将其中的一个分区设置为活动分区

在ROM中，紧挨着检测主引导扇区指令的是一个跳转指令，所以，一旦主引导记录被读入内存，中央处理器将从ROM那里跳到这里接着执行。主引导扇区的启动指令将分析读入内存的分区表，看看哪个是活动分区，接着，算出该分区的起始位置，从那个扇区读入操作系统写在那里的引导代码，也将它读入内存，然后接着执行。通常，操作系统的引导代码位于各分区前若干个连续的扇区内，这一切都安排的很好，一环扣一环，就这样一步步操作系统 把自己最重要的部分全部读入内存中

从主引导扇区开始，操作系统 把自己读入内存并开始执行的过程称为“自举”，尽管谁也不能把自己举起来，但这里是指令的世界，物理上的定律不起作用。在这里，主引导扇区承担了一个承上启下的作用，基于这个原理，要是自己编写一段代码放在主引导扇区里，不管用来做什么，都一定会非常有趣

从开机到操作系统启动完毕，多数情况下是很顺利的，但如果不走运的话，也会出问题。麻烦在于，产生根源可能来自许多可能的方面，有硬件方面的原因，有操作系统系统的因素

## 15.4 操作系统的功能

为了让用户方便地操作计算机，操作系统 承担了繁重的工作，不过对于普通用户来说，唯一需要关心的是能在屏幕上看到硬盘里都有什么东西，有哪些软件程序可以运行，以及如何让它们开始运行

硬盘上的东西分两种：一种是软件程序，它们将在合适的时候被中央处理器执行一下；另一种尽管也包含了二进制数据，但它们本质上不是处理器可识别的指令，而是代表别的意义，是软件处理的原始材料或加工的结果，比如MP3音乐歌曲、用字处理软件编辑的通讯录和方案等。无论程序或数据，用户最关心的还是如何在硬盘上看到它们

要讨好用户满足他们的愿望，操作系统有大量的工作要做，这些工作通常不为人知，而我们也不需要知道，因为这正是操作系统的本分——掩盖这些细节，不劳我们亲自动手

几乎没有例外，我们每人人在购买计算机后做的第一件事情就是为它安装操作系统

因为有了操作系统，我们可以浏览磁盘上的文件，如果某个文件是可执行的，用鼠标双击一下它，当然这期间会发生很多事情，不但要操作系统把它装入内存，还要中央处理器的参与

根据我们平时使用计算机的经验，像Windows这样的操作系统可以同时运行很多程序，比如一边用音乐播放器听歌，一边写文章，同时还开着聊天程序

这很奇怪吗？当然，对于一台普通的个人计算机来说，可能只有一个中央处理器，这样的话一次只有一个程序单独在内存中运行，完事后再进行下一个，但，现在看起来所有的程序，包括操作系统都在同时运行中

几乎所有的计算机都拥有中断，在主板上，有一个时钟电路，每隔一段时间就会向中央处理器发出一个中断信号。中断通常是无条件产生的，不管中央处理器当时在干什么，它应当放下当前的任务，去处理中断，而它所执行的中断指令恰恰是操作系统的一个组成部分。这是操作系统自己有意安排的。这部分程序将检查当前内存中的所有任务，看看下一个该轮到谁，然后跳到它上一次被中断的地方接着运行。这样，所有的程序都在计算机内轮流执行，直到完成自己的工作。在此期间，没有哪个程序会始终独占中央处理器

对于常规的任务来说，比如在计算机上写文章，或浏览网页，处理器的负担并不重。随着处理器的速度越来越快，这种工作方式使得处理器的闲置率越来越高，当你在屏幕上编辑文字时，处理器仅仅是在等待

一个字符，而在等待的这一小会儿，它应该被充分利用。基于这个原因，现代的操作系统必须允许同时运行多个程序，而对于大型计算机来说，甚至还应当允许很多用户通过各自的终端同时在它上面工作，这就是操作系统的多用户、多任务特性。多用户和多任务是依靠处理器和操作系统共同完成的，处理器提供像中断这样的硬件支持，操作系统在软件上予以配合和扩展。当然，弄巧成拙的时候也是有的，当你发现鼠标不灵了，或者播放的音乐断断续续时，说明这种工作机制出现了问题。

操作系统的多用户、多任务能力不是最近才有的。事实上，自然电子计算机出现之后不久就有了，在那个时候，内存容量很小，人们当然希望同时运行的程序越多越好，但一个很大的矛盾是当时的技术无法提供足够的内存来支持这种需求。

20世纪90年代，我们可以使用的内存才几兆，就是这样还让我们大小写兴奋。1971年的时候，个人计算机使用内存只有几千字节。

尽管随着技术的进步，个人计算机上的内存容量一直在不断增加，但对内存的需求增加的更快，实际上是快得多。你想想，就那么一点点地方，操作系统要占很大一部分，毕竟它是大管家，另外，各种硬件设备的驱动程序要占一部分，剩下的那些空间还要被各个用户程序瓜分。多数的计算机用户通常是不满足于同时只做一件事情的，他们会打开聊天程序、听着音乐，顺便还打开网页写写今天的心情，还要玩会游戏。内存中的程序越来越多，编写程序的人把程序写的太庞大，尽管现在的个人计算机可以有几GB字节，但这样消耗内存很快会不够用的。

怎样才能使有限的内存空间服务于越来越多同时运行的程序呢？一个有效的办法是：请考虑一个场景，内存中有许多程序在同时运行，但同一时间只能有一个获得处理器的光顾，如果内存空间不够的话，可以先使那些不用的数据和代码（这些数据和代码属于当前程序或其他程序）退避到外部磁盘，腾出空间来加载将要使用的这部分。当那些被移到外部磁盘的东西又要被使用时，再用相同的方法调入内存。这样，程序员在写程序时，不必为是否有足够的内存空间而担忧；而对于正在运行程序来说，它们会觉得自己正在一个巨大的存储空间中运行。

这听起来像是一个骗术，是的，但你只说对了一半，关键是，它真的很有效。这种方法最早产生于20世纪70年代，通过它，使得任何程序都可以有巨大的内存空间供自己使用，虽然插在主板上的内存条实际上只有很小的容量，由于这种方法变出来的内存并不存在，而是位于磁盘上，所以称为虚拟内存。

像处理器在访问高速缓存时因为不命中而受到惩罚一样，因为磁盘是慢速设备，比内存慢很多，所以当需要访问的内容不在内存中，而要从磁盘上载入时，代价是巨大的。下次当你感觉计算机反应很慢，而硬

盘指示灯也在不停地闪烁时，一定要记得那是操作系统正在内存和硬盘之间倒换数据，换句话说，虚拟内存技术是以牺牲程序执行时间为代价的，特别是当你同时运行了太多的程序时

在我们平时使用的软件中，操作系统无疑是最复杂的，当然，它包含的指令数量也是最多的，它管理磁盘文件，允许我们执行程序，让一大堆程序在内存中工作，当它们保姆，为它们提供虚拟内存。在它的绝活中，还有一个未曾提到，那就是设备管理功能，当你用电线电缆把各种各样的设备连接到主机上时，有没有想到，如果没有操作系统，我们的软件将难以同这些设备建立友好的关系。举例来说，当内存中所有程序都想在屏幕上显示自己的内容时，该怎么办呢？再比如，对于一个大的办公室来说，所有的计算机通过网络共用一台打印机是一种比较经济的办法，在这种情况下，如何才能使所有要打印文件的人可以简单地发出打印命令，而不需要焦急地坐在那里等着别人的打印任务完成呢？换句话说，应该有专门的方法来管理打印机、接受打印任务并使它们排队

编写一个操作系统是一个庞大的工程，而要完全了解它的工作也不是这区区几页纸所能办到的。感兴趣的读者可以选择系统地学习指令系统课程

## 第16章 办公、娱乐和程序设计

在对计算机的原理有了充分认识后，不得不说，计算机的确和其他器物不一样，计算机的用途大得多、广泛得多。

如果说计算机的历史，不得不说布莱士·帕斯卡，他是17世纪法国的数学家、物理学家、哲学家，他曾经发明过一种计算机（当然是机械的），同时为了纪念他在科学上的贡献，还有一种以他的名字命名的计算机语言。帕斯卡说过：“我们人类的全部尊严就在于思想，”，他还说“人只不过是一根芦苇，是自然界最脆弱的东西，但他是一根有思想的芦苇。”他的这番话道出了思想对于我们作为“人”这种生物的重要性。相似地，软件其实就是计算机的思想，根据我们已掌握的知识来看，毫无疑问，计算机的能力来自硬件和计算机指令相结合的功劳。当在大堆计算机指令组合在一起，用于实现各种各样的目的时，就成了我们所说的计算机软件。

现实中有各种各样的软件，这是很自然的，因为我们有许多事情要让计算机帮我们处理。新的软件不断产生，所以我们不可能逐一认识每个软件，只能粗略地挑几个重点的来说一说。在这个过程中，可能需要将它们分成几类



## 16.1 用于编辑文章和排版的文字处理软件

计算机发明之后不久，人们就使它具有了文字处理功能，而且是最受欢迎、使用最多的功能之一

为了在计算机上编辑文字，需要一种特别的软件，通常称为文本编辑软件或文字编辑软件，高级一些的称为文字处理软件或文字处理系统。最简单的文字编辑软件就是Windows内置的“记事本”软件，对于Unix操作系统来说，是一个叫vi的软件。

记事本的功能很简单，不能设置单个字符的颜色和大小、不能调整字间距和行间距，也不能插入图片。记事本软件的这些限制和它在磁盘上的存储方式有关，记事本把文字从第一个字开始按顺序一个个的代码写入磁盘扇区，直到最后一个字。

在这种情况下，文件内容仅仅是所有字符的代码，不包括其他额外的东西，比如每个字的大小和文字。以后任何时候，当再次查看这个文件时也只能得到和显示这些文字，至于文字的大小和颜色，文件中没有记录，所以我们把这种仅仅包括文字代码的文件称为纯文本文件，或者文本文件。

和文本文件不同，为了追求更好的文字处理效果，需要使用更高级、功能更强大的软件，比如WPS（*Word Process System*，文字处理系统）。

和记事本软件不同，WPS即可以处理纯文本文件，也可以处理非文本文件，即带格式的文件。所谓带格式的文件，是指文字内容包括多种属性，如字体形状、文字大小、颜色、加粗、加下划线等等。同时，可以调整字间距和行间距，选择文字内容的对齐方式，也可以在文字中插入各种图片

为了在以后准确地还原这些文字内容以及它们的属性和格式，在磁盘上保存一个纯文本文件是不行的。在这种新的文件中，不单要保存文字本身，还要保存文字属性，事实上，不同的文字处理软件会采用不同的存储方式。这意味着文件和生成它的软件基本上是一一对应的，别的软件可能会无法识别它，除非知道它的存储方式。现在应该可以体会纯文本文件的价值，因为它仅仅包含文字内容，所以具有通用性

WPS在国内的市场占有率一度达到95%，就在WPS如日中天时，美国的比尔·盖茨带领人员开发Windows操作系统，这是一个图形化界面的操作系统，人们用鼠标点击来取代繁琐的键盘操作，而在此之前，人们的计算机屏幕上只能显示常规的字符和简单的图案，Windows一经推出就获得了成功，在此基础上，微软公司紧接着开发出了能在Windows下运行的字处理软件Word，人们喜欢Windows，Word能运行在Windows

下，而WPS则不能，WPS受到了重创，从1993年开始市场份额逐年下滑。

不过从1997的开始，求伯君和他的WPS再度崛起，现在WPS不但拥有Word的全部功能，而且更本土化。

## 16.2 压缩和解压缩

文字处理软件是我们平时用得最多的软件之一，称为办公软件。当然，它其实也只是众多的办公软件之一。不管是在办公室还是在家里，对于我们这些离不开计算机的人来说，发送电子邮件、把文件上传到网上或从网上下载文件都避免不了，网速是有限制的，要是文件太大或者文件太多，我们会想到将它们变小一些，以节约传输时间，对方接收到文件再将它恢复原样，这个过程，称为在压缩/解压缩

压缩和解压缩不是最近才出现的需求，事实上，它很早就有了。在没有因特网，甚至连U盘也没有出现的时候，人们习惯于用软件在计算机之间复制文件，典型的软盘容量1.44MB，当然很小了。要最大限度地利用软盘的空间来保存更多的文件，压缩和解压缩就很关键了

通过压缩可以减小文件的大小，比如，原本一个5MB的图片，经过压缩后可以变成1MB，压缩率是20%。注意，通常情况下，压缩必须是无损的，换句话说，必须保证压缩/解压缩过程中恢复的数据与原始数据完全相同。想想看如果一个可执行文件解压缩之后有一些字节发生了变化，那么就意味着指令的改变，指令的改变势必使程序发生不可预知的错误。再比如，解压缩之后的金融数据被改变了，同样会导致严重的后果，既然是这样，为什么一个文件能够被压缩？这是什么道理呢？

首先，不管是什么文件，它的数据都是大量的二进制比特，它们以字节为单位存放，至于这些二进制数据代表什么含义，取决于其生成者和使用者之间协议

其次，在很多时候，这些二进制数据可以用另一种更简短的形式来表示，这时，称为这些数据是可压缩的，比如，一个最容易理解的例子是行程编码。

行程编码的思想是观察被压缩的数据，看它是否带有重复的内容，假如我们要压缩这样一串数据：

AAAAAFFRRRU

那么，因为我们发现有些字母是连续重复出现的，所以这串数据可以简单地表示为：

5A2F3R1U

注意，计算机只接受二进制比特，所以AAAAAFFRRRU在存储器和磁盘中其实是这样的：

01000001 01000001 01000001 01000001 01000001 01000110 01000110  
01010010 01010010 01010010 01010101

按上面所讲的行程编码方法，压缩后的结果是这样的：

00000101 01000001 00000010 01000110 00000011 01010010 00000001  
01010101

可以看出，压缩之后显然节省了若干个字节，如果一个软件知道这个文件是用行程编码压缩的，它就懂得如何恢复原来的内容

很显然，并不是所有的文件都可以被压缩，尤其是不存在重复内容的时候，压缩之后反而会增加数据量，不过，行程编码方法对于压缩图像文件特别有效，因为图像是由像素组成的，而像素则是一些表示颜色的二进制数据，最关键的是，图像中总是包含大量具有相同颜色的像素

行程编码并不是唯一的文件压缩方法，但，不管有多少种方法，基本的思想没有改变，那就是为被压缩的内容找一种更简短、更节省空间的表示方法

## 16.3 图像、音乐和视频

软件很多，可以根据它的应用范围分成多种类别，每种软件都有各自的特点和使用方法

在过去若干年里，数字图像的使用增加了，增加的原因是显示器的发明和显示技术的飞速发展，图像就是图像，之所以称为“数字图像”是因为在计算机内部图像不过是大量的二进制数据

在计算机可以接受图像之前的年代里，图像只意味着一支画笔或一架装有胶片的照相机或摄像机，但要让图像进入计算机，就需要使用将图像变在二进制数据的设备，如数码照相机、数字摄像机、扫描仪、摄像头.....

对图像数字化有三个好处：第一，最新的存储技术可以保证图像不容易被破坏，可以无限制地复制，存储成本也较小；第二，可以在个人计算机、智能手机这类设备上随时浏览和欣赏；第三，不像传统的纸张和胶片，可以通过软件来改变图像背后的二进制数据，从而实现对图像的编辑

编辑图像需要借助于专业的图像编辑软件，通过这种工具，可以改变图像原有的效果，甚至可以得到自然界从来没有过的图像，如果需要，可以把一个人脸上的痣去掉、修饰体形、改变肌肤颜色.....这一切可以做得天衣无缝

近年来，半导体图像设备和图像编辑软件的发展造就了一些新的职业，有很多人活跃在印刷、户外广告设计、产品包装设计、因特网网站设计等领域，所有这些通常称为平面设计

图像文件的主要内容是像素数据，也就是一系列的二进制字节，代表着图像上每个点的颜色，以一个24位真彩色图像为例，每个像素点需要使用3字节来表示（分别是它的红、绿、蓝三原色调配比例）。如果图像的分辨率是1280\*1024，那么这幅图像在存储器中需要占用的空间是：

$$1280*1024*3=3932160\text{字节}=3.75\text{MB}$$

现在来看，3.75MB好像算不了什么，但数量多了，会占用大量磁盘空间。而且更早以前，因特网的速度还不是很快，但图片在网页中的应用却很广泛，所以即使是现在，太大图片对我们的耐心和因特网带宽（带宽也称吞吐量，是指一段特定时间内（通常是1秒）网络所能传送的比特数。例如，如果每秒能传送1000万个比特，就称此时的网络带宽为10Mbps）都是个考验

为了减小图像的体积，可以采用前面的无损压缩技术，但由于它的原则是无损，所以并不能保证总是有效，特别是对那些“含水量”不是太

多的复杂图像。在这种情况下，1986年，国际标准化组织和国际电话电报咨询委员会等几个组织，共同组成了一个致力于改善图像压缩方法的联合图像专家小组（Joint Photographic Experts Group, JPEG），“联合”的意思的恰如其分地指出了该小组的成员来自各个组织，同时也表明了这是一个各方联合的成果，1992年，表决通过了该标准的第一部分

JPEG的研究既涉及对人体视觉特点的分析，还要借助于复杂的数学公式，一般说来，每一幅图像都具有一个总体的色调和基本特征，以及一些反映局部细节的微小变化。

基于这个原理，JPEG将图像分成 $8 \times 8$ 块，然后一块一块地对图像进行处理。首先，第一个环节是分析图像，它的核心思想是找出那些观看图像所必需的总体特征和那些不太必要的、在有些情况下人眼几乎感觉不出来的细微特征。直白地说，这一步的动作是决定哪些图像信息是可以删除的

接着，JPEG使用精心选取的64个数作为除数，一对一地将上一步输出的结果截短（做了除法之后当然会变小）。这64个数也是按 $8 \times 8$ 排列的，称为量化表。根据对图像压缩品质的要求，JPEG准备了好几套量化表，这就好比是准备了好几个粗细不同的筛子，要想理解量化的过程，可以想象你准备压缩100以内的几个数，比如，43，25，69，81和7。可以将它们统统除以10分别得到几个近似的结果4，2，6，8，0。这样，就可以用4个比特来保存它们，而不是原样保存的时所需的1字节。当然不同的是，JPEG的压缩使用量化表，而不是单个固定的数字。在这个过程中，很多对分辨图像来说不太重要的部分都变成了0

最后，也是第三步，要对量化后的数据按传统的方法进行压缩，量化过程是有损失的，但第一步和这一步则没有

通过选择不同的量化表和其他一些参数，就能控制压缩率和逼真度。一般公认的是JPEG能以30:1的比率压缩24位真彩色图像。注意这是一个多元化的世界，而JPEG也不是唯一的图像压缩，所以还能看到GIF、TIFF、PNG这类图像格式

考虑到人们对电影和电视的偏爱，除了编辑和浏览图像，计算机也可以播放和编辑视频。事实上，观看视频已经成为人们最主要的娱乐项目

在电影院里，运动的图像是由单个的电影胶片一张一张快速显像而形成的，利用的是人眼的视觉暂留特点，在计算机上，活动的视频也是由单个的图像在屏幕上快速切换显示而成的，这些单独的图像称为帧

麻烦的是，如果用未经压缩的图像来拼凑一部完整的电影，将占用大量存储空间，而且，以目前的因特网带宽，这对于那些喜欢在网上看电影的朋友来说绝对不是一个好消息

为了压缩或播放（解压）视频，同样需要一个标准，没有标准，别人可能不知道如何播放你制作的视频节目。为了压缩视频，这一回又成立了一个新的小组，即我们前面介绍过的运动图像专家组（MPEG），MPEG不仅定义了如何压缩视频的方法，也定义了压缩音频的标准，比如现在人尽皆知的MPEG Layer III（MP3）

为了压缩视频，MPEG可采用JPEG的方法来压缩视频中的每一帧，因为运动图像就是以某种速度连续显示的静止图像。但总是只解决了一半，还不能到此为止。为了得到稳定的画面，电影的播放速度一般是每秒钟24帧，要在计算机上播放，可能还要高一些，尽管帧是经过压缩的，但数量很大，仅仅压缩静止的帧远远不够

不过，考虑一下，如果视频中没有许多活动，比如一个变化不大的面部特定，两个连续的帧会包含几乎相同的内容，即使视频是活动的，很多时候也只是它在屏幕上的位置发生了变化，上一帧到下一帧也会有大量的冗余内容，认识到这一点，MPEG还必须删除帧之间的冗余部分

这意味着，经过MPEG压缩后，组成视频的不再是传统意义上的静止图像那么简单了。事实上，这些帧分三种：第一种是参考帧，它是独立的，理论上包含完整的画面信息；后面两种类型的帧是不完整的，不能独立存在，要依赖于前面的帧、后面的帧或者参考帧。最后，MPEG的压缩比例可达150：1，一般来说，能达到90：1已经很不错了

根据不同的应用场合及对图像清晰度和声音效果的不同要求，MPEG标准实际上分好几个层次，比如，MPEG-1是VCD使用的编码标准，而现阶段比较流行的DVD光盘则采用的是MPEG-2，仅视觉效果上看，DVD显然比VDC要清楚得多

MPEG也不是唯一的视频编码标准，比如，国际电信聪明电脑标准部（ITU-T）定义了H系列的标准H.261和H.263，它们与MPEG的工作很相似，但在细节上有所区别

计算机硬件的发展，再加上显示技术越来越先进，越来越逼真，大大促进了广播电视、电影和动画产业的数字化进程，以前，所有的电视台都用录像带来记录和播放电视节目，录像带用电磁感应原理记录声音和图像。最近若干年，这一切都数字化了，图像和伴音不再是模拟的，而是被编码成二进制比特，促进人们进行这种转变的原因是因为计算机只能处理二进制数据，在把图像和伴音数字化之后，就可以把这些电视节目拿到功能强劲的计算机上，对它们进行字幕叠加、淡入淡出、画中画等特技处理。如果没有数字技术，不会在电视上看到那么多特殊效果的画面

传统的动画制作非常麻烦的，需要一张一张地作图，每一幅图和上一幅图有一点点细微的差别，当它们连续播放时，就看到了连贯的动

作，这就是利用人类视觉暂留特点制作和播放动画的过程

以前制作一部动画片是很累人的事情，要画大量的图像，现在可以用特殊软件在计算机上进行，甚至通过给软件输入一些参数，就可以控制细微的面部表情和肢体动作，达到非常逼真的效果



## 16.4 计算机语言和编译软件

20世纪90年代，有人说要想精通计算机知识，非得掌握计算机原理、汇编语言和数据库和计算机网络。

数据库和计算机网络是相对专业的知识。这里我们可以简短地讨论一下包括汇编在内的计算机语言，来看看今天数量庞大的计算机软件是怎样生产出来的

对于很多人来说，再没有比能够让计算机根据自己的指令来完成一些神奇的事情更有意思的了。计算机之所以有用，关键在于它能够运行程序

为了编写程序，历史上曾用过形状和纸带，这两种方法有一个共同的特点，那就是直接使用二进制来工作，换句话说，计算机说什么话我们也得跟它说一样的话，目的是希望它能听懂。

注意上面最后一句话很有意思，就是把计算机看成一个人，而它执行的指令就是语言，尽管这只是一个比方，但很显然，“计算机语言”已经成了一个流行的词汇。

二进制地计算机来说是最直接的，我们可以用中央处理器的指令集来直接编写程序，这些指令称为机器语言，比如下面的机器语言指令片段，它们位于某台计算机的存储器中，可以运行在我们平时使用的计算机上，目的是从该计算上的第一块硬盘里读主引导扇区，并将它们写入指令中指定的内存地址

```
10111000 00000001 00000010 10111011 00000000 00000010 10111001
00000001 00000000 10111010 10000000 00000000 11001101 00010011
00000000 00000000
```

把这种东西称为机器语言是有道理的，因为它是机器的工作语言，只有机器认得它。它是计算机的“官方语言”，但对人类来说，这一串串0和1很容易让人头晕，即使是最熟练的工程技术人员也不能保证不会出错

在这种情况下，编程人员开始尝试像平时说话一样编写程序，比如，计算55除以11可以写成

```
mov ra, 55
```

```
div ra, 11
```

我们前面接触过。当时我们将这些作为计算机指令的助记形式，现在看来，它们可以作为一种计算机语言。

和以往不同，这种新的计算机语言（汇编语言）提高了编写程序的效率，同时也将人们从构造机器指令的泥潭中拯救出来了，用更多的时间和精力专注于解决问题的方法和思路。

注意，用形状或纸带直接向内存中写入机器指令的方法是几十年前就已经被淘汰了，所以不要想着如何去亲自做一下，体验一下直接向内存中写指令是多么神奇。现在为了编写汇编语言程序，需要一个文本编辑器，比如Windows下的记事本或Unix操作系统下的vi软件，当通过键盘输入所有的汇编语言指令后，保存为纯文本文件。纯文本是任何软件都能直接处理的文件格式，只包含纯粹的字符代码。

基本上，汇编语言的每一行语句都和一条二进制的机器指令相对应，所以可以将它看成是符号化的机器语言。保存在硬盘上的汇编语言文件，包含的是用键盘输入的字母和数字，也是一连串的0和1，但都是键盘字符的编码，和实际的机器指令相去甚远，中央处理器不会认得它们，比如，把立即数55装载到寄存器RA的机器指令可能是这样的：

11001001 00110111

而它对应的汇编语言

mov ra, 55

在内存和磁盘文件中则是一串字符代码

01101101 01101111 01110110 00100000 01110010 01100001 00101100  
00110101 00110101

很明显，需要有一种方法把汇编语言转换成二进制的机器语言

用汇编语言写出来的东西称为汇编语言源程序。为了把汇编语言源程序编码成机器语言指令，需要使用编译程序。这种程序和其他所有程序一样，唯一不同的是它不用来写文章，听音乐……它本身就是程序，却被用来生成更多的其他程序，真是不可思议

编译程序（有时称为编码器）不是操作系统的组成部分，很多时候需要从专业的软件公司购买，然后通过操作系统安装到计算机硬盘，并接受操作系统的统一管理。在此之后，就可以像使用任何其他软件一样来用它工作，有时，某些编译软件自己也会带有文本编辑器，可以在它里面像编辑文章一样一行一行地编写汇编语言语句，然后命令它将这些内容翻译成机器语言并保存到一个程序文件中。

任何一种计算机语言都有自己的规则和限制，比如用哪些单词代表什么指令，每条指令按什么格式书写，等等。比如，不能将

mov ra, 55

写成

move ra, 55

mov ra. 55

mov ra, mm

对我们来说，一眼就看出mov是连在一起的单词，但计算机不行，要把汇编语言转变成机器指令，编译器要做的第一件事是打开源程序文件，

一字节一字节地取出文件的内容，判断它们，将连续的字符组装成单词，如上面的`mov, ra, 55`，在组装的过程中，如果遇到 空格、逗号、回车键等，就意味着当前单词组装完毕，应该组装下一个单词了

组装完毕后，编译器还要进一步分析这些单词是不是它所期望的，比如，如果把`mov`写成了`move`，编译器一定不认识，所以只好停止编译，并在屏幕上显示一些错误信息。即使它认识`mov`也还要进一步分析它后面的内容是不是合乎语法，按道理，`mov`指令的第一个参数必须是寄存器或存储单元，所以如果要所一条汇编语言指令写成

`mov 55, ra`

那么编译器会指出这个错误，并停止编译

对于那些熟悉计算机内部结构的人来说，使用汇编语言是一件十分惬意的事情，一方面它比直接书写二进制直观，另一方面，在书写每条语句时，仿佛可以看到中央处理器内部繁忙的场景，他们像熟练的管道工一样，随心所欲地把水从这里引向那里，很有成就感。

除了机器指令外，汇编语言无疑是最有效率的计算机语言，想想看，它直接操纵计算机内部的每一个微小部分，操控每一个工作细节，还有什么比它更了解计算机核心呢？

但很奇怪，汇编语言居然一直被认为是低级语言，这么强大的东西，怎么会被认为是低级的呢？当然，这个“低级”不是一个贬义词，实际上，它指的是接近硬件的程度和层次，所以正确的理解应当是，汇编语言是一种低层次的语言很靠近硬件，同时，它也晦涩难懂，不是吗？

尽管汇编语言在易书写和易理解方面前进了一大步，但仍有很多不便之处，别的不说，要用汇编语言编写程序，必须了解计算机的内部构造和工作细节，有很多人对计算机硬件一知半解，但也想写程序来解决手头的问题，对他们来说，这个门槛无疑太高了，在这种情况下，人们希望计算机语言能远离硬件，最好像平时说话一样简单，这就是高级语言

发展高级语言的目标之一是让我们不必考虑内存地址，这意味着，像汇编语言一样，需要用字符或字符串来代表某个内存单元，比如，可以用

`int i`

表示一个内存位置，`int`的意思是要用这个内存位置来存放一个整数，`i`是该内存位置的代号，可以认为它是一个内存地址，但具体是内存中的哪个地方，不用操心，编译器和操作系统会自动处理这些事情。`int`是关键字，是固定的，高级语言依靠关键字来推测和识别我们的意思。至于`i`可以随意，完全可以用`j`, `num`, `int1`任意字符来取代它

所有东西在内存中都是二进制，不管它们代表着什么 - 整数、小

数、键盘字符、有多少袋大米.....但对于多数高级语言来说，在你指定一个内存单元的同时，还必须指定它的用途 – 存放整数、小数？字符、一串字符（字符串）？如此等等，它这样做完全是为了我们好，当我们把一个整数和字符串相加时，它就可以提出抗议并拒绝继续编译，因为这没有意义，性质不同

一旦有了一个可用的内存位置*i*，就可以读/写它了，或者用它来运行各种算术逻辑运算。比如，可以将一个整数写入该内存位置

*i*=1250

紧接着，还可以用*i*里面的内容与另一个数进行计算，比如

*i*=*i*\*30

这条语句的意思是，用*i*原先的内容乘以30，结果再写入*i*。因为计算机键盘上没有乘号，所以通常用星号来代替，几乎所有的高级语言都是这样

对于*i*来说，它其实就是临时的内存单元，用于存放运算的中间结果，既然有了*i*，我们紧接着肯定会用它来读读写写，以致于它的内容会随着程序的运行而发生改变。由于这个原因，像它这样的东西在高级语言里称为变量

很显然，高级语言的另一个好处是不必关心计算机的内部构造，包括那些神秘的寄存器，任何一个人，即使不知道计算机的内部构造，不知道都有哪些寄存器可用，也照样能写出程序。我们曾用汇编语言来做1-100的加法，现在我们用C语言来做同样的事情：

```
int i=0;
int s=0;
while(i<=100)
{
s=s+i;
i=i+1;
}
```

在C语言中，任何语句都以分号结尾，这是惯例。第1行和第2行说明了两个变量*i*和*s*。注意，在说明的同时可以直接赋值，这是允许的，程序按顺序执行，执行到这里将开辟两个内存位置，并写入指定的数据（在这里都是0）

和*int*一样，C语言中，*while*是关键字，不能拼错。它专用于重复做某些事情，意思是“在符合.....条件的情况下，重复做.....事情”

重复做一件事得能够结束，不能永远不停，这就是*while*的作用，它后面跟着一个判断条件，在这里，判断条件是*i*是否小于等于100，和乘法一样，因为键盘上没有≤，所以用<=代替

要重复做的事情跟在while后面，用大括号括起来，首先，用s的内容加上i当前的内容，结果仍保存在s中，接着i在原来的基础上加1

所以总的来说，s和i最初都是0，循环刚开始时， $i=0$ ， $s=0$ ，当 $i=1$ 时， $s=0+1$ ，当 $i=2$ 时， $s=1+2$ ，当 $i=3$ 时， $s=3+3$ .....就这样不断重复，直到i大于100停止

顺便说一句，尽管用高级语言编程不涉及寄存器这些东西，但那只是高级语言为你掩盖了它们

C语言不是最早的编程语言，但它是最受欢迎的编程语言。C起源于1965年一种叫BCPL的语言，意思是“编译器编写和系统编程工具”。在BCPL的基础上，后来产生了B和New B，接着New B变成了C，时间在1971年和1972年之间，这个名字既说明了它们之间的继承关系，同时也表明了它的发明者希望它真的简捷有效

在本章的开头，我们曾提到过布莱士·帕斯卡，以及用他的名字命名的计算机语言。帕斯卡1623年生于法国，在数学和物理学上有突出贡献，他发现了随着高度降低，大气压强会增大的规律，还和别人一起测量了同一地点的大气压变化情况，开创了利用气压计进行天气预报先河。后来，人们就用他的名字作为压强的单位“帕（Pa）”

1641年，为了减轻他父亲（税务官）的工作负担，帕斯卡制造出世界上第一台手摇计算机，可以计算六位数的加减法。1971年，当瑞士苏黎世联邦工业大学的沃斯教授发明了一种新的计算机语言，为了纪念帕斯卡的伟大发明，他将这种计算机语言命名为帕斯卡（Pascal），在很长一段时期内，PASCAL语言在高校计算机软件教学中一直处于主导地位。

用高级语言来编写程序有几个明显的好处：

第一，容易阅读。人类的记忆力并不完美，时间一长，连自己最初为什么要写这些程序代码都会忘记，使用高级语言可以缓解这样的症状，同时也可以让其他同事理解和使用这些代码

第二，现实世界中有各种类型的计算机，它们使用不同的微处理器，连操作系统都可能不一样。要在这些计算机上完成相同的工作，必须分别为它们编写不同的机器指令或汇编语言程序，原因很简单，不同的计算机有各自不同的指令系统。但，如果使用和硬件无关的高级语言，可以避免这样的处境。只需要针对某一个高级语言，为每一种类型的计算机设计出一个编译器，就能把同一个高级语言程序翻译成多种不同的机器指令，并运行在每一种不同类型的计算机上，这就是所谓的跨平台和可移植性

第三，也是最重要的一点，高级语言提高了软件编写的效率，极大回忆了程序编写的速度。高级语言的一条语句相当于几条乃至几

十条、几百条机器指令，甚至，高级语言提供了大量现成的功能，可以像搭积木一样拿过来就用，而不必重复编写。使用高级语言提供的功能，只需要简单的语句就可以完成工作，即使它需要成千上万条的机器指令

不过，高级语言虽然有千般好，也有些缺点。比如编译后生成的可执行文件都比较大，原因很简单，编译器没有类似于人的智慧，它总是以最可靠、最全面的方式来编译程序，不免会做一些画蛇添足的事情。你能想象自己半夜三更起床去厨房找吃的也要衣着整齐吗？可事实上，编译器总爱做这样的事情，所以，如果你的程序对反应速度比较敏感（比如要在极短的时间里精确控制车床），就不要使用高级语言了，应该选择汇编语言

除此之外，尽管高级语言可以掩盖计算机内部的细节，让人们不是十分了解计算机原理的情况下书写程序，但达成这个目标的代价是引入另外一些复杂的东西，甚至不比学习硬件知识容易。

## 16.5 计算机病毒也是软件

在过去几十年里出现了大量的软件，尽管具体数量不清楚，但至少是一个巨大的数字。大多数软件在设计时都遵循一个原则，那就是它要按用户的要求来工作

比如只有在你按退格键时，光标所在那个字符才会被删除；只有当你按回车键时，文字处理软件才会另起一行；同样地，如果不是你亲自单击了“关机”命令，计算机决不会自作主张把电源断掉（停电或计算机硬件故障除外）

然而，就像菜刀可以切菜，也可以伤人一样，如果一个人不怀好意，他也可以编写专门搞破坏的软件，比如，硬盘引导扇区通常只会在操作系统安装的过程中被修改，而这种修改是必需的，用户也是认可的，否则就无法启动计算机。但有些人就偷偷摸摸地在别人的计算机上安装并运行一些程序，私自毁坏主引导扇区的内容，以至于下次无法正常开机

再比如，删除文件通常是需要我们亲手操作的，但这些破坏者会偷偷地删除文件，造成数据丢失。编写这种软件的人通常怀有各种不同的目的，而这种软件被称为计算机病毒。

计算机病毒的编写者通常都有一个共同的特点，就是希望这种恶意程序能够通过磁盘和计算机网络到处传播，破坏的计算机越多越好，所以这种软件都会包含传播和向其他计算机上复制的指令代码，基于自然界里的病毒也具备这些特征，所以把这种程序称为“计算机病毒”非常恰当

为了安全地使用计算机，对计算机进行相应的调整并安装防病毒软件是非常必要的。

# Table of Contents

- [第1章 了解计算机，要从电开始](#)
- [第2章 用电来表示数](#)
- [第3章 怎样才能让机器做加法](#)
- [第4章 电子计算机发明的前夜](#)
- [第5章 从逻辑学到逻辑电路](#)
- [第6章 加法机的诞生](#)
- [第7章 会变魔术的触发器](#)
- [第8章 学生时代的走马灯](#)
- [第9章 计算机时候的开路先锋](#)
- [第10章 用机器做一连串的加法](#)
- [第11章 全自动加法计算机](#)
- [第12章 现代的通用计算机](#)
- [第13章 集成电路时代](#)
- [第14章 核心与外部设备间的接口](#)
- [第15章 计算机的启动过程和操作系统](#)
- [第16章 办公、娱乐和程序设计](#)